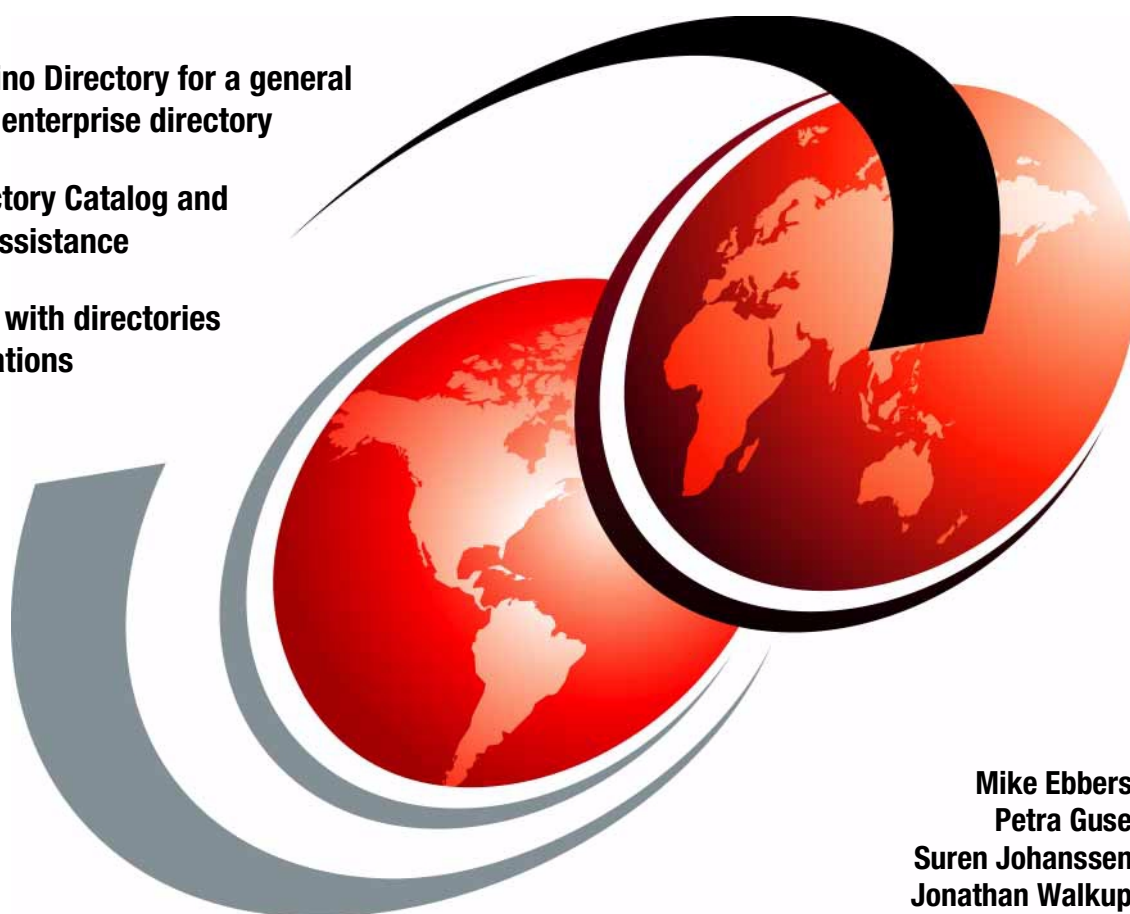


# Getting the Most From Your Domino Directory

Using Domino Directory for a general purpose or enterprise directory

Using Directory Catalog and Directory Assistance

Integrating with directories and applications



Mike Ebbers  
Petra Guse  
Suren Johanssen  
Jonathan Walkup

[ibm.com/redbooks](http://ibm.com/redbooks)

# Redbooks





International Technical Support Organization

## **Getting the Most From Your Domino Directory**

**November 2000**

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix I, "Special notices" on page 263.

**First Edition (November 2000)**

This edition applies to Lotus Domino Release 5.0.4

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. TQH 1CP-5605E  
1 Charles Park  
Cambridge, Massachusetts 02142-1245

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

**© Copyright International Business Machines Corporation 2000. All rights reserved.**

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

## Contents

<b>Figures</b> .....	ix
<b>Tables</b> .....	xiii
<b>Preface</b> .....	xv
The Team that Wrote this Document .....	xv
Comments welcome .....	xvii
<b>Chapter 1. Introduction</b> .....	1
1.1 Purpose .....	1
1.1.1 History of the Domino Directory .....	1
1.1.2 The value of a general-purpose enterprise directory .....	2
1.2 The Lotus/IBM approach to directories .....	3
1.2.1 Critical infrastructure .....	3
1.2.2 Standards-based .....	4
1.2.3 Ongoing investment .....	4
1.3 Road map to this book .....	4
<b>Chapter 2. Domino Directory services</b> .....	5
2.1 The role of directory services in Domino .....	6
2.2 The Domino Directory .....	7
2.3 Directory Catalog .....	9
2.3.1 User Directory Catalog .....	10
2.3.2 Server Directory Catalog .....	11
2.3.3 Directory Catalog configuration example .....	12
2.4 Directory Assistance .....	12
2.5 Directory searches .....	14
2.5.1 Directory search order .....	15
2.5.2 Settings affecting the search sequence .....	16
2.5.3 Search sequence for address verification in an LDAP directory .....	18
2.5.4 Search sequence for authentication for a Web client .....	19
2.5.5 Search sequence for group authentication .....	21
2.6 Authentication and security services .....	21
2.6.1 Authentication for applications via NRPC .....	23
2.6.2 Authentication for LDAP clients .....	27
2.6.3 Authentication in the Domino Directory services .....	29
2.7 Domino Directory as an enterprise directory .....	30
<b>Chapter 3. Planning your directory services</b> .....	33
3.1 What are the requirements and business drivers? .....	33
3.2 Discovering your organization .....	34

3.3	Defining directory ownership and administration . . . . .	37
3.4	Directory aggregation, synchronization and publication . . . . .	38
3.5	Planning your Domino Directory services . . . . .	39
3.5.1	Is this a new deployment of a Domino infrastructure? . . . . .	40
3.5.2	Are you enhancing or expanding your directory services? . . . . .	40
3.5.3	Are you deploying Domino as an enterprise directory? . . . . .	41
3.5.4	The Domino Directory services details . . . . .	42
<b>Chapter 4.</b>	<b>Setting up and configuring Domino Directory services . . . . .</b>	<b>47</b>
4.1	Installing the Domino server . . . . .	47
4.2	Setting up and configuring LDAP services on Domino . . . . .	48
4.2.1	Setting up the LDAP service on an existing Domino server . . . . .	48
4.2.2	Configuring the LDAP service . . . . .	49
4.2.3	Exporting the LDAP schema . . . . .	52
4.2.4	Modifying the LDAP Schema . . . . .	53
4.3	Setting up and configuring Directory Assistance . . . . .	56
4.3.1	Setting up rules for a Domino Directory . . . . .	56
4.3.2	Setting up rules for an LDAP directory . . . . .	57
4.3.3	Deploying Directory Assistance . . . . .	59
4.4	Setting up and configuring Directory Catalogs . . . . .	59
4.4.1	Setting up a server Directory Catalog . . . . .	60
4.4.2	Configuring the Directory Catalog . . . . .	60
4.4.3	Running the Directory Catalog aggregator . . . . .	64
4.4.4	Scheduling the Directory Catalog aggregator . . . . .	65
4.4.5	Making the Directory Catalog available to the server . . . . .	66
4.4.6	Setting up and configuring a mobile Directory Catalog . . . . .	66
4.4.7	Deploying a mobile Directory Catalog to end users . . . . .	66
<b>Chapter 5.</b>	<b>Administering Domino Directory services . . . . .</b>	<b>69</b>
5.1	Monitoring Domino Directory services . . . . .	69
5.1.1	Monitoring the Directory Catalog . . . . .	69
5.1.2	Monitoring the LDAP service . . . . .	71
5.2	Performance tuning Domino Directory services . . . . .	75
5.2.1	Performance tuning for Notes users . . . . .	75
5.2.2	Performance tuning for LDAP users . . . . .	75
5.2.3	Performance tuning for address lookup . . . . .	76
5.2.4	Performance tuning for authentication . . . . .	76
5.3	Troubleshooting Domino Directory services . . . . .	77
5.3.1	Troubleshooting the Directory Catalog . . . . .	77
5.3.2	Troubleshooting Directory Assistance . . . . .	79
5.3.3	Troubleshooting LDAP lookups against external directories . . . . .	79
5.3.4	Troubleshooting LDAP lookups against Domino Directory . . . . .	79
5.3.5	Troubleshooting authentication issues with external directories . . . . .	80

5.4 Guidelines and best practices for using the Domino Directory . . . . .	81
5.4.1 Take an enterprise-wide approach . . . . .	81
5.4.2 Managing multiple Domino directories . . . . .	83
5.4.3 Designing an efficient Domino Directory . . . . .	83
5.4.4 Streamlining administrative processes . . . . .	83
5.4.5 Modifying the standard Domino Directory design . . . . .	85
5.4.6 Security considerations for a Domino Directory . . . . .	85
5.4.7 Leveraging Directory Catalog and Directory Assistance . . . . .	86
5.4.8 Leveraging the inherent benefits of the .nsf format . . . . .	87
5.4.9 Enforcing schema checking . . . . .	87
<b>Chapter 6. Using Domino Directory services . . . . .</b>	<b>89</b>
6.1 Client access to Domino Directory services . . . . .	89
6.1.1 Domino Directory services for the Notes client . . . . .	89
6.1.2 Differences for Internet clients . . . . .	98
6.2 Using standards-based tools for directory manipulation . . . . .	100
6.2.1 LDIF . . . . .	100
6.2.2 LDAPsearch . . . . .	103
6.2.3 Other command line tools . . . . .	104
6.2.4 LDAPSsync . . . . .	104
6.3 Using application development tools with Domino Directory services	105
6.3.1 Using the Notes API . . . . .	105
6.3.2 Using the LDAP C API . . . . .	106
6.3.3 Other programming tools . . . . .	107
<b>Chapter 7. Directory integration . . . . .</b>	<b>109</b>
7.1 Overview . . . . .	110
7.2 Integrating with other directories and applications . . . . .	110
7.2.1 IBM and Lotus products . . . . .	110
7.2.2 Microsoft products . . . . .	116
7.2.3 iPlanet (Netscape/Sun) products . . . . .	134
7.2.4 Novell products . . . . .	140
7.2.5 Other LDAP directories . . . . .	143
7.2.6 Metadirectory products . . . . .	144
7.2.7 Others . . . . .	156
<b>Appendix A. Industry groups . . . . .</b>	<b>157</b>
A.1 The Open Group (TOG) . . . . .	157
A.2 Directory Interoperability Forum (DIF) . . . . .	157
A.3 Distributed Management Task Force (DMTF) . . . . .	157
A.4 Directory Services Markup Language (DSML) . . . . .	158
<b>Appendix B. LDAP and X.500 Standards . . . . .</b>	<b>159</b>
B.1 LDAPv3 . . . . .	159

B.2 LDAPv3 RFCs . . . . .	160
B.2.1 Core Specifications . . . . .	160
B.2.2 Extended Core RFCs . . . . .	161
B.2.3 Other Related RFCs . . . . .	162
B.3 Internet Draft -- LDAP Extensions . . . . .	167
B.3.1 Applicability and Review of LDAPv3 . . . . .	167
B.3.2 LDAP Controls & Operations . . . . .	169
B.3.3 CLDAP . . . . .	174
B.3.4 Other Extensions . . . . .	175
B.3.5 Access Control, Authentication & Authorization . . . . .	176
B.3.6 Sorting and paged retrieval of search results . . . . .	178
B.3.7 Directory-Enabled Networking . . . . .	179
B.3.8 LDAP APIs . . . . .	180
B.3.9 Synchronization . . . . .	182
B.3.10 Replication (LDUP/LCUP) . . . . .	182
B.3.11 Internet Directory Services . . . . .	185
<b>Appendix C. Domino Directory forms . . . . .</b>	<b>189</b>
C.1 Certificates . . . . .	189
C.1.1 Notes certificates . . . . .	189
C.1.2 Notes cross certificates . . . . .	190
C.1.3 Internet certificates . . . . .	191
C.1.4 Internet cross certificates . . . . .	191
C.2 Configuration settings . . . . .	192
C.3 Connection documents . . . . .	193
C.4 Domain documents . . . . .	193
C.5 External domain network information . . . . .	193
C.6 Group documents . . . . .	194
C.7 Holiday documents . . . . .	195
C.8 Location documents . . . . .	195
C.9 Mail-in database documents . . . . .	195
C.10 Person documents . . . . .	196
C.11 Program documents . . . . .	197
C.12 Resource documents . . . . .	198
C.13 Server documents . . . . .	198
C.14 User setup profile documents . . . . .	199
<b>Appendix D. Syntax of LDAPSearch command . . . . .</b>	<b>201</b>
<b>Appendix E. Description of test environment . . . . .</b>	<b>203</b>
E.1 balder.lotus.com . . . . .	203
E.2 odin.lotus.com . . . . .	203
E.3 heimdal.lotus.com . . . . .	204
E.4 gefion.lotus.com . . . . .	204



<b>Appendix F. Sample code using the Notes API</b>	205
<b>Appendix G. Basic directory concepts</b>	217
G.1 What is a directory?	217
G.1.1 Directory clients and servers	218
G.1.2 Distributed Directories	220
G.1.3 Directory security	221
G.1.4 Users, platforms, and networks	222
G.1.5 Directory versus database	223
G.1.6 Directory synchronization	225
G.2 Directory standards	228
G.2.1 X.500 - the Directory Service Standard	228
G.2.2 X.509 certificates	237
G.2.3 LDAP	239
G.3 Enterprise directory	244
G.4 Metadirectory	246
G.4.1 Metadirectory Systems	247
G.4.2 Metadirectory Product Architecture	248
G.5 Emerging Trends	249
G.5.1 Converging on standards -- LDAP	249
G.5.2 Leveraging directory services -- Directory Enabled Networks	251
G.5.3 Common data definition -- DSML	251
<b>Appendix H. Directory Schema</b>	253
H.1 Attributes	256
H.2 Object classes and mapping to LDAP schema	259
<b>Appendix I. Special notices</b>	263
<b>Appendix J. Related publications</b>	267
J.1 IBM Redbooks	267
J.2 IBM Redbooks collections	268
J.3 Other resources	269
J.4 Referenced Web sites	269
<b>How to get IBM Redbooks</b>	271
IBM Redbooks fax order form	272
<b>Index</b>	273
<b>IBM Redbooks review</b>	279



---

## Figures

1. Enabling exhaustive lookup . . . . .	18
2. Authentication process flow . . . . .	25
3. Lookup sequence in the authentication process . . . . .	30
4. Domino server setup with LDAP selected . . . . .	48
5. LDAP configuration settings. . . . .	50
6. Modifying fields available to anonymous users . . . . .	51
7. Document in Schema50.nsf. . . . .	52
8. Directory Catalog configuration: basics . . . . .	61
9. Directory Catalog configuration: Advanced . . . . .	63
10. Directory Catalog configuration in the server document . . . . .	65
11. Setup profile form. . . . .	67
12. Directory Catalog statistics report . . . . .	70
13. TCP server probe form . . . . .	74
14. Address picker dialogue. . . . .	90
15. Display of available directories . . . . .	91
16. Quick find locating an entry . . . . .	92
17. Directories displayed below the searchable line . . . . .	93
18. Search panel provides multiple options . . . . .	94
19. Ambiguous Name dialogue resulting from pressing F9 . . . . .	96
20. LDIF import dialog . . . . .	101
21. SecureWay default referral setup . . . . .	111
22. SecureWay directory with referral entries . . . . .	112
23. SecureWay Directory Assistance configuration . . . . .	113
24. ACL for database with SecureWay group listed . . . . .	114
25. Overview: LDAPSync solution . . . . .	115
26. Installing Domino Directory NT Sync Services . . . . .	117
27. Menu options added to NT User Manager. . . . .	117
28. Notes registration setup . . . . .	118
29. Notes Mail/ID registration options . . . . .	119
30. Notes deletion and user synch options . . . . .	120
31. Notes registration dialog . . . . .	120
32. Add user to Windows NT from Domino Administrator . . . . .	121
33. Migrating Windows NT users and groups . . . . .	122
34. Windows NT user name parsing options . . . . .	123
35. Notes - NT Single Logon install . . . . .	124
36. Directory Assistance configuration for Active Directory . . . . .	125
37. ACL for Active Directory users. . . . .	126
38. Directory Assistance configuration for the Exchange 5.5 server . . . . .	127
39. Exchange 5.5 LDAP referral setup . . . . .	128
40. Outlook LDAP directory service setup . . . . .	129

41. Exchange user migration dialog . . . . .	130
42. Adding Domino extension mapping to IIS . . . . .	131
43. Installing the Domino ISAPI filter in IIS . . . . .	132
44. Directory Assistance configuration for iPlanet LDAP server . . . . .	134
45. Domino ACL with an entry from an iPlanet server . . . . .	135
46. iPlanet Directory Server Referrals configuration . . . . .	136
47. Adding a Smart Referral to an iPlanet directory entry . . . . .	137
48. External LDAP directory configuration for iPlanet Web server . . . . .	138
49. IPlanet Web server ACL using an external directory . . . . .	139
50. Importing an LDIF file into iPlanet directory server . . . . .	140
51. Directory Assistance document for eDirectory server . . . . .	141
52. Group of eDirectory users . . . . .	141
53. ACL of database for eDirectory users . . . . .	142
54. Referral setup in eDirectory . . . . .	143
55. Directory registry before synchronization . . . . .	145
56. Selecting the MMS server . . . . .	146
57. Creating a management agent . . . . .	146
58. Beginning to configure the management agent . . . . .	147
59. Completing the configuration of a management agent . . . . .	148
60. Updating server information . . . . .	149
61. Master information . . . . .	150
62. Management agent log . . . . .	151
63. Updating the metadirectory . . . . .	152
64. Adding links . . . . .	152
65. HTTP logon screen . . . . .	153
66. Password prompt . . . . .	153
67. A successful logon . . . . .	154
68. Updating a metadirectory entry . . . . .	154
69. Confirmation of update . . . . .	155
70. Directory client/server interaction . . . . .	219
71. Directory synchronization . . . . .	226
72. Distributed directory model . . . . .	230
73. Directory entry structure . . . . .	233
74. Example of a directory information tree . . . . .	233
75. Referrals a . . . . .	235
76. Referrals b . . . . .	235
77. Uni-chaining . . . . .	236
78. Multi-chaining . . . . .	237
79. Mixed modes hybrid approach . . . . .	237
80. LDAP server acting as a gateway to an X.500 server . . . . .	243
81. Stand-alone LDAP server . . . . .	244
82. A metadirectory architecture . . . . .	248
83. An enterprise metadirectory . . . . .	249

84. LDAP object classes in Domino R5.02 .....	256
85. DominoGroup structural object class .....	259
86. DominoPerson structural object class .....	260
87. \$PersonGeneralInfo auxiliary object class .....	261



---

## Tables

1. Directory Catalog vs. Directory Assistance . . . . .	20
2. Levels of user access to a database . . . . .	26
3. Types of access. . . . .	28
4. Field to schema mapping. . . . .	54
5. LDAP statistic definitions . . . . .	72
6. LDAPSearch parameters. . . . .	201
7. Example ACL for an employee's directory entry . . . . .	222
8. Typical user credentials . . . . .	227
9. Domino OID (object identifier) allocation . . . . .	255
10. General attributes . . . . .	256
11. Personal attributes. . . . .	257
12. Organizational attributes . . . . .	258
13. Security . . . . .	258
14. Ancillary . . . . .	259





---

## Preface

From the very beginning, Domino Directory, originally known as the Name and Address Book (NAB), has been a key part of the Domino architecture. It has evolved from being specific to Lotus Notes and Domino to serving as a general purpose directory. This redbook explains Domino Directory services, how to plan for and implement them, and how they can be extended to work with other directory services.

We start by discussing the purpose of this redbook and the IBM-Lotus strategy of directories. Then we explore the wide range of services that Domino Directory offers. We also discuss access to Domino Directory from an application development angle through protocols and technologies such as Notes API, LDAP, and JNDI.

In addition, we look at technical approaches to directory integration and also a few real world examples where Domino Directory integrates with other directories. Finally, we cover some good practices in designing a directory infrastructure.

This redbook is written for IT architects, Domino administrators, and other technical professionals involved with directory structures

---

## The Team that Wrote this Document

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Cambridge Center.

**Mike Ebberts** is a senior IT specialist at the International Technical Support Organization, Cambridge Center. He writes extensively and teaches IBM classes worldwide on Domino for S/390. Mike has been with IBM for 26 years. Before joining the ITSO six years ago, he worked for 10 years as a systems engineer and 10 years as an instructor/developer.

**Petra Guse** is a Senior Consultant at Lotus Professional Services in the region Frankfurt, Germany. She has four years of experience in messaging and migration environments, specially with infrastructure planning and installation in the Lotus and Softswitch product area. Before this she worked more than six years in user and network support in different heterogeneous environments.

**Suren Johanssen** is a Lotus Domino architect for IBM Global Services in North Harbour in the United Kingdom. He has seven years of experience with Lotus Notes and Domino. He has worked at IBM for four years in the network computing environment. His areas of expertise include infrastructure planning and enterprise deployment. He is currently part of a Notes infrastructure team that maintains over eight hundred servers and a user base of 120,000.

**Jonathan Walkup** is a Senior Notes Administrator for Symbiosys, Inc. of Rockville, MD. He is currently on assignment at The World Bank in Washington, DC. Jon has been using Notes since Release 2 and administering Notes/Domino systems since Release 3. He is a Principal Certified Lotus Professional in both Application Development and System Administration, for both R4 and R5.

Thanks to the following people for their invaluable contributions to this project:

David Morrison  
International Technical Support Organization, Cambridge Center

Soren Peter Nielsen  
International Technical Support Organization, Cambridge Center

Keith Attenborough  
Lotus Development Corporation, Westford

David Goodman  
Lotus Development Corporation, Scotland

Parastoo Vakili  
Lotus Development Corporation, Mountain View

Boris Vishnevsky  
Lotus Professional Services, Wayne

In addition, we would like to thank the following people:

- Peggy Bovaird, IBM, Somers
- Scott M Davidson, Iris
- Mike Halliday, IBM, Watson
- Neil Hawkins, IBM, North Harbour
- Beth Keach, Iris
- Patrick Lin, Iris
- Dave Martin, IBM, Raleigh

- Mike O'Brien, Iris
- Jon Reinke, IBM, Almaden

---

## **Comments welcome**

### **Your comments are important to us!**

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "IBM Redbooks review" on page 279 to the fax number shown on the form.
- Use the online evaluation form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)



---

## Chapter 1. Introduction

At the heart of Lotus Notes and Domino is Domino Directory. It is a fundamental part of the architecture that delivers Domino's scalability, security and standards-based flexibility. Yet the use of Domino Directory is so transparent that many people don't realize the power of this directory architecture.

---

### 1.1 Purpose

The purpose of this redbook is to act as a guide to help you understand and get the most from the directory services provided by Lotus Domino Release 5. It is designed to act as both an executive-level overview of the directory capabilities included in Domino and as a pragmatic guide to designing and implementing an enterprise directory solution.

Lotus Domino Release 5 provides a broad range of Domino-specific and standards-based directory services. These services combine to greatly simplify user and systems management, act as a container for a consistent set of data across an organization, and support Web-based and distributed applications across the corporate network. Domino Directory services also provide support for Notes/Domino integrated public key infrastructure (PKI) as well as third party PKI deployments, such as those provided by Entrust Technologies and RSA Security.

Domino Directory services have evolved throughout the years, with continuous improvements and innovations, so that Domino Release 5 provides quite a flexible directory infrastructure. This evolution enables existing customers to continue to grow and exploit Domino Directory as their general-purpose directory even as their enterprise grows in size and sophistication. The evolution also enables Domino and Domino-based applications to fit smoothly into the multi-directory environment present in many organizations today.

#### 1.1.1 History of the Domino Directory

The directory is not a new technology area for Lotus. They have always recognized the importance of a centrally held, universally accessible store for information describing users, groups, and other resources in the enterprise. From the first release in 1988, Lotus has incorporated a directory as an integral part of its Notes and Domino products. Initially called the "Name and Address Book", the directory in Notes and Domino has continued to evolve over the last twelve years to fully embrace all the features and functions now

associated with enterprise directory services. Now the Domino Directory has matured to meet the increasing demand for robust and scalable enterprise directory services.

As the industry and enterprises have more recently begun to recognize the practical importance of directory services, pragmatic standards have been developed to assist in providing interoperability between different directory structures. The leading set of standards gaining widespread commercial acceptance in recent years has been the Lightweight Directory Access Protocol (LDAP). Lotus has been an early and consistent proponent of the adoption of these emerging standards, and introduced LDAP v2 into the Domino Directory services in Release 4.6. With Release 5, Domino Directory services have incorporated all major LDAP v3 functionality and continue to encompass additional standards as they evolve.

In addition to leveraging the power of standards-based technologies, Domino Directory services also build on the power of Lotus Domino technology to provide significant value not yet widely available across the community, including strong and finely granular access controls, robust multi-master replication technology, and cross-platform support that has been a hallmark of Notes and Domino technology.

One of the key consistent characteristics of Domino Directory services is that they have been accessible not only to users and administrators, but also to application developers. A variety of applications (in addition to Notes and Domino) utilize information stored in the Domino Directory through a wide variety of interfaces, from the well-documented and widely published Notes and Domino API set to third-party LDAP development toolkits. The history of Notes and Domino development has been consistently one of innovation, striving towards openness, which has had a major impact on the directory services.

This pattern of innovation carries forward into Release 5 with the evolution of Domino Directory Assistance and the introduction of the Domino Directory Catalog. Both of these components of the Domino Directory services are described in additional detail later in this redbook.

### **1.1.2 The value of a general-purpose enterprise directory**

The value of an integrated enterprise directory environment is easy to see. It crosses the three major communities of administrators, developers, and users.

First, administrators have a coherent structure with which to work. This minimizes their involvement with information transfer between directories, and there are fewer administration tools to learn and employ. This simplification reduces total cost of administration.

Second, application developers have access to a single consistent and well-defined source of enterprise information. This information can include not only data about users but also internal and external security and policy data, as well as information on other resources held by the enterprise. By having such an authoritative and consistent source, application development is greatly simplified. There is the potential for significant code reuse and a reduced need for repetitive data validation. The enterprise can benefit by having a set of applications that are directory-independent. This provides the added flexibility of selecting their directory vendors while avoiding being locked into a single vendor solution.

Third, users benefit from access to more reliable and consistent data about people, groups, and resources across the enterprise. At its simplest, the availability of a single authoritative source of internal phone numbers across IBM has been a tremendous time saver. At the more complex end of the scale, having a single consistent profile of key customers and business partners will be the foundation for e-Business.

The remainder of this redbook will provide insight into how to plan and implement a Domino Directory infrastructure that will help you reap practical benefits. Lotus has a history of innovating and implementing directory technology for more customers than any other company in the market.

---

## **1.2 The Lotus/IBM approach to directories**

With the rising popularity of directories, people are asking about the philosophy and products of IBM and Lotus in this area.

### **1.2.1 Critical infrastructure**

Both Lotus and IBM are committed to providing strong, standards-based directory technology to our customers, including robust directory offerings such as the Lotus Domino Directory and the IBM SecureWay Directory. Lotus/IBM products also support the ability of applications and middleware (hosted on application platforms such as Domino and WebSphere) to use directory technology to integrate smoothly with and leverage other directory products in the enterprise. IBM and Lotus' focus on providing this dual capability is based on an underlying belief that directories are infrastructure components that should be ubiquitous and interoperable. Furthermore, once

directories achieve this fully interoperable status, a significant constraint on the deployment of e-Business solutions will be eliminated.

### **1.2.2 Standards-based**

Consistent with this unifying principle, both Lotus and IBM are actively involved in the development of standards for directories, including LDAP. In addition to participating in the working groups of the IETF, Lotus and IBM are involved in the Directory Interoperability Forum (DIF), the Distributed Management Task Force, the Directory Services Markup Language initiative, and the Open Group.

### **1.2.3 Ongoing investment**

Lotus' directory technology investments support these underlying beliefs. The Lotus Domino Release 5 Directory is fully LDAP v3 compliant, providing a solid platform for a central enterprise directory. Ongoing investments focus on continuous improvement in scalability and performance to meet the growing demands of large enterprises. Extensions of the Directory Catalog and Directory Assistance technologies will also address these needs, as well as assist in smoothly integrating Domino applications and platforms into an enterprise with a multi-directory environment. Implementation of key emerging standards such as Access Control for LDAP, as well as ongoing extensions to schema and naming flexibility will ensure that Domino remains compliant with key standards as they are implemented.

---

## **1.3 Road map to this book**

For readers who would like a refresher of directory concepts and terminology, we provide Appendix G, "Basic directory concepts" on page 217. The other chapters assume familiarity with the material in Appendix G.

In Chapter 2 we discuss how basic directory concepts and standards are represented in the Domino Directory model. Chapter 3 can serve as a planning guide for a directory project that involves Domino. Chapter 4 shows how to set up and configure the Domino functionality described in Chapter 2. Chapter 5 gives administrators responsible for Domino Directory servers some practical advice. Chapter 6 describes how to access the Domino Directory through standards-based products and how to write applications that can take advantage of the Domino Directory. Finally, Chapter 7 discusses how to integrate the Domino Directory with other directory services, and provides two detailed case studies to illustrate the point.



---

## Chapter 2. Domino Directory services

The directory architecture in Domino Release 5 has evolved dramatically, to the point where it is now more appropriate to refer to it as “Domino Directory services”, of which the Domino Directory is just one (although the central) component. When used to its full extent, the Domino Directory architecture provides a robust capability equally capable of acting as either the center of an enterprise directory infrastructure or as a peer directory in a multi-directory environment.

The other components of the Domino Directory services include:

- Directory Catalog -- an aggregation of directories located on either the server or client
- Directory Assistance -- provides access to federated directories that can include secondary Domino directories or third-party LDAP directories
- Domino LDAP server task -- running on the Domino server and providing LDAP version 3-compliant access to Domino and third-party directories for both clients and applications

Each of these components is discussed in detail in this chapter.

### Note

If you would like to review the concepts and terminology of the Domino Directory environment, refer to Appendix G, “Basic directory concepts” on page 217.

The terms “aggregation of directories” and “federation of directories” deserve fuller explanation before moving on:

- *Aggregating* directories is the process of selecting specific objects and attributes from a number of specified directories in an organization and collecting them in a new consolidated source that can be independently distributed and searched. In Domino Directory services, the Directory Catalog represents such an aggregation of information from secondary Domino directories.
- *Federating* directories is the process of allowing directory information to remain in its original directory or location and providing a mechanism to redirect (or refer) queries for that information to those original directories. The information is joined in a more “federal” style, retaining its original form and location, but appearing as one to an outsider. In Domino Directory services, the Directory Assistance provides this “federating”

service for both secondary Domino directories and third-party LDAP directories.

In the rest of this chapter we discuss the Domino Directory architecture in detail. As an integral part of directory services, we discuss authentication and security services. Finally, we discuss the opportunity for using the Domino Directory as an enterprise directory.

---

## 2.1 The role of directory services in Domino

The Domino Directory services consist of a primary Domino Directory and any combination of the following components:

- One or more secondary Domino directories
- A server Directory Catalog
- One or more user Directory Catalogs
- A Directory Assistance database
- The Domino LDAP service

The following roles and functionalities are supported by the Domino Directory services:

- The primary Domino Directory as a domain configuration store and centralized point of domain management

All certificates, connections, cross certificates, server configurations and domain documents are maintained in the primary Domino Directory. This enables easy domain administration as the administrators just need to update the details in the Domino Directory in a single Domino Directory in the domain. Replication will then ensure the details are pushed to all the servers in the domain.

- Mail address lookup and resolution service using the Domino Directory services

Type-ahead functionality and name resolution are provided by any combination of primary and secondary Domino directories, server and user Directory Catalogs, and Directory Assistance, which include LDAP directories for referrals. The solution is focused on easier and quicker mail addressing. Depending on the scale of the solution this may also include dedicated directory servers to allow load balancing and a minimized impact on normal mail servers.

- User authentication and authorization using the Domino Directory services

The default Notes user authentication procedure via NRPC is certificate-based. It is also possible to switch off certificate-based authentication for Notes users, which will enable Notes users to remain anonymous until the user attempts to perform an operation that is not allowed to be performed by anonymous users. For non-Notes clients there are three levels of security and authentication: anonymous access, name and password authentication, and certificate-based authentication. For more information see 2.6, “Authentication and security services” on page 21.

- Domino Directory services as a user information store and central point of information management

Depending on the requirements, this may be more than just managing the primary Domino Directory as a centralized information store. In the primary Domino Directory, this will include management of people and group information, and it can also include resource information. If Domino is used as the hub in an enterprise directory solution, this may also include using Domino as the centralized point for managing secondary directories.

- Referral services to other directories from the primary LDAP directory

You use Directory Assistance to refer LDAP Lotus Notes clients that connect to the Domino LDAP service to another LDAP directory. This only happens if the search in the Domino LDAP service's primary Domino Directory and all secondary Domino directories configured in Directory Assistance isn't successful. For more information on the referral services, see 2.4, “Directory Assistance” on page 12.

Domino's directory services represent a unique solution for each deployment or environment, as each organization will use a concatenation of Domino Directories, Directory Catalogs and Directory Assistance with one or multiple LDAP directories, according to their needs.

---

## 2.2 The Domino Directory

The Domino Directory is the central store for directory information used by the Domino servers and by clients accessing applications and services hosted on those servers within a Domino domain. The first server in a domain creates the first instance of the Domino Directory as it is initially installed. Each subsequent server in the same domain creates a replica copy of this same directory as part of its installation process. These copies will be kept synchronized through periodic replication across the domain, providing a robust and distributed directory architecture.

The information maintained in this primary domain directory can be broadly divided into two categories -- information associated with the configuration of the Domino servers, the domain they reside in, the services they provide and how they interrelate, and information about the people and groups who access those servers and use those services and applications provided. The configuration information is primarily of interest to Domino administrators, who will maintain and update it to ensure smooth operation of the environment. The people/group information is of much wider interest, and can be made accessible to a wide range of users, developers and applications. These applications can be either hosted on Domino servers or can be legacy applications hosted on a wide variety of platforms and systems.

From the perspective of the Domino server and applications, both types of information are stored in a series of documents, which are organized in views and displayed using forms, with individual items of information contained in fields. For a list of the document types, how to configure and use them as well as some real world examples, see Appendix C, "Domino Directory forms" on page 189. The folders and views are provided in a format that makes sense to the users of the directory. The normal user will use the people and groups views only. The rest of the default views, including hidden views, are used by administrators and the Domino server.

**Important**

Do not make any changes to the hidden views in the Domino Directory. They are needed in their default format by the Domino server and clients.

**Tip**

Be very careful when adding additional views to the Domino Directory. Every view adds to the size of the directory and workload to each server, because each server has to maintain the view indexes.

As an example, the IBM UK address book has a default size of 398 MB with 33,000 registered users. The default "People" view is 15 MB in size and the \$Users view is 71 MB. So you can see how quickly the directory can grow with additional views.

The Domino Directory is also an LDAP v3-accessible directory store. This means this same information can be viewed as being held in an LDAP schema and consists of a series of object classes and attributes accessible via LDAP search and manageable through LDAP add, delete, modify and

rename operations and LDAP-compliant tools. The Domino LDAP schema includes standard schema elements, such as Person, OrgPerson, and inetOrgPerson, and can be extended using the Domino designer. The schema can be published using either LDIF commands or into a predesignated Domino database.

---

## 2.3 Directory Catalog

The Domino Directory Catalog is a specialized database populated with entries from one or more Domino directories. The information contained in the Directory Catalog is controlled by the Domino administrator using the configuration document in the database. The entries are created by the Directory Catalog task, a server task running in the background on the Domino server. The Directory Catalog (DirCat) task can be scheduled or run manually.

The primary purpose of the Directory Catalog is to provide a lightweight, quick access store of directory information primarily for use by mobile and disconnected users. It achieves this goal through a combination of three specific methodologies. These include selection of the information to be included, aggregation of the information into consolidated documents, and elimination of most indexed views.

- Selection of information

The creator of the database controls the information included in the Directory Catalog in three ways, using the Directory Catalog configuration document contained in the Directory Catalog itself:

- By specifying the individual secondary Domino Directories that will be used as the sources for the Directory Catalog. While we recommend that these be replicas of the secondary directories that are available locally, they can be hosted by remote servers.
- By selecting the individual fields that will be aggregated into the Directory Catalog. The Fullname and Listname fields are always present, and a suggested list of fields is proposed, but the database creator can add and delete items from this list.
- By selecting the records that are included. The Advanced tab of the configuration document provides for definition of a selection formula.

- Aggregation of information

Space is saved by aggregating multiple entries from the source directories into a single document in the Directory Catalog, using the "Packing Density" specified in the configuration document as a guide. The default

(and maximum) value is 255, meaning that each individual storage document can hold up to 255 individual entries. This saves significant overhead in document storage. There are other implications, discussed later in this redbook, of this aggregation approach.

- Elimination of most indexed views

Indexed views provide a valuable service by significantly speeding up search operations, however, they also consume significant space. The Directory Catalog contains the absolute minimum number of views needed to fulfill its functions. The database creator selects one of three predefined sort orders (Full Name, Last Name or Alternate Full Name) to be the primary sort order for the entries. Queries, such as type ahead, that are presented in that order use the indexed view to locate responses. So, if the database creator selects Last Name as the sort order and the user begins typing an address in by last name then first name (Smith J...), then the indexed view will be used to locate the entry quickly. If information is entered in a different order, then the search will be handled by the full text search engine.

To be more specific, a Directory Catalog has three small hidden views, one visible view and a “virtual” view, as follows:

- The \$Users view contains the aggregate documents and is used for name lookups.
- The \$Unid view contains information needed by the dircat task to replicate the secondary directory entries to the Directory Catalog. This view does not get created on replicas of the Directory Catalog, which further reduces the Directory Catalog size. It does not get built on the user’s workstation when the user replicates the user Directory Catalog.
- The \$PeopleGroupsFlat view is used to display directory names when Notes users click the Address button to browse directories.
- The visible view “Configuration” shows the document used to configure the Directory Catalog.
- The virtual view “Users” is the view that users and programs can open and access to see the names included in the Directory Catalog. This view is created on the fly, as needed.

There are two different types of Directory Catalogs that you can deploy, user Directory Catalogs and server Directory Catalogs. They both have specific roles in the Domino Directory services.

### **2.3.1 User Directory Catalog**

This is also called a mobile Directory Catalog. The functionality and advantages of the user Directory Catalog are:

- Notes users use local replicas of a user (mobile) Directory Catalog to enable them to do quick mail addressing to anyone in the organization, even in disconnected mode.
- Users can use encrypted mail in disconnected mode. When sending an encrypted memo, it is flagged for encryption. At the next server connection, when the Notes client sends the mail item, the client looks up the public key and encrypts the mail on the fly. This is referred to as “just in time” or “on the fly” encryption.
- Group names can be included in the catalog, so users can address mail to groups. When mail is sent, the group lookup will happen on the server and the router will include all the group users in the recipient list.
- By using the LDAP protocol, users can search in the Directory Catalog the same way that they search a personal address book.
- Users can use the address assistant dialog box to open and scroll through the names in the Directory Catalog.
- Network traffic is reduced because most of the name resolution occurs locally on the workstation, rather than on a server.
- If the soundex is enabled and you are not sure how to spell a person’s name or surname, you can just guess it as close as possible. The Directory Catalog will work through all the entries and will return a list of possible matches.

### **2.3.2 Server Directory Catalog**

The functionality and advantages of the server Directory Catalog are:

- Notes users can do very quick name searches in secondary Domino directories on the server.
- If a server Directory Catalog is enabled on the user’s mail server, Notes clients without user Directory Catalogs can use the server Directory Catalog to address mail and browse directory entries.
- The mail router can look up addresses more quickly in a server Directory Catalog rather than using Directory Assistance to look up the addresses in multiple, individual secondary Domino directories.
- If you set up Directory Assistance, the LDAP service can use the Directory Catalog and Directory Assistance together to process LDAP searches, providing the functionality of both.
- The server Directory Catalog is also used in the Web client authentication process. If the client exists in a Directory Catalog and not in the primary Domino Directory, the server uses the information available through Directory Assistance to very quickly access the complete entry in the secondary Domino Directory. This is possible because each entry in the Directory Catalog includes the replica ID of the Domino Directory from

which the entry was derived, and the unique ID associated with a replicated document.

- The server Directory Catalog can be the central point for all organization level directory access, for applications, LDAP clients and other components in the Domino environment.
- If the soundex is enabled and you are not sure how to spell a person's name or surname, you can just guess it as closely as possible. The Directory Catalog will work through all the entries and return a list of possible matches.
- When the mail router uses the Directory Catalog, it performs an exhaustive lookup of all entries in the Directory Catalog, even if the "Exhaustive lookup" router configuration option is disabled, because the router can do the exhaustive lookup quickly in one database.

### **2.3.3 Directory Catalog configuration example**

The Domino administrator can configure the Directory Catalogs to suit the organization's needs. A typical setup will be for an organization to use more than one Directory Catalog. The configurations of a user Directory Catalog and a server Directory Catalog will be slightly different because of their different roles. To find out more about how to configure the Directory Catalog, see 4.4.2, "Configuring the Directory Catalog" on page 60.

As an example, the IBM user Directory Catalog is sorted by firstname because the default usage in the organization for mail addressing is the first name, surname format. This approach limits the amount of network traffic sent to the mail server as most of the address resolution will be done on the workstation. The IBM server Directory Catalog is sorted by lastname and includes more fields, as well as group information, which will resolve type-ahead entries entered with the surname, name format if the user's "mail file location" field in the active location document is set to "on server".

---

## **2.4 Directory Assistance**

Domino Directory Assistance is the third major component of Domino Directory services. Its primary purpose is to provide the mechanism to federate one or more secondary directories, making them transparently accessible to directory users. These secondary directories can be either Domino Directories or third-party LDAP-compliant directories such as Netscape's iPlanet or Novell's NDS. Directory Assistance supports the following functionality:

- Finds entries in secondary Domino directories and LDAP directories on behalf of Notes users for mail addressing.



- Finds entries in secondary Domino directories on behalf of LDAP clients.
- Refers LDAP clients to LDAP directories.
- Uses name and password security to authenticate web clients registered in secondary Domino directories.
- Uses x.509 certificates to authenticate Web clients registered in secondary Domino directories.
- Authenticates Web clients registered in LDAP directories.
- Expands groups for authorization of Web users in a single selected LDAP directory.
- Provides failover to another replica of a secondary Domino Directory.
- Uses naming rules to efficiently search secondary Domino directories.
- Provides support for "Recipient name type ahead" addressing.
- Works in conjunction with a Directory Catalog on a server.

Directory Assistance provides the administrator with a great deal of flexibility in establishing parameters for the use of the federated directories. These controls include determining which of the LDAP directories will be used for group expansion for authentication, identifying name matching rules which can direct a search to a particular directory for resolution, determining which directories can be trusted to provide legitimate credentials for authentication processing, providing its own credentials for performing authenticated binds to secondary directories, and other capabilities. All of these characteristics are determined by settings on Directory Assistance documents maintained in a Directory Assistance database on the Domino server.

When discussing Directory Assistance, there are two terms that need some explanation. They are very similar, but have definite differences. They are defined in RFC2251 paragraphs 4.1.11 and 4.5.3:

- A *referral* is returned when the target host does not contain the requested entry itself, but does have knowledge of a host that may contain that entry. The search result returned to the client contains one or more referrals to that other directory or directories. The client then uses the referral to look in the new target directory.
  - To return a referral, the Domino LDAP service never connects to and searches the LDAP directory.
  - Instead, the service uses information in the Directory Assistance document to return a referral.
  - As a result, the client only needs to make one call to either the master or the replica.
  - The referral includes the URL host name for the LDAP directory server, the base distinguished name configured for the directory, and the port the LDAP directory uses.

- A *continuation reference* is returned when the target host holds some, but not all, of the subtree that is targeted by the search request and has knowledge of a host that may contain additional results. In this case, the search result returned to the client contains that portion of the response the original host does hold and a continuation reference to the other directory or directories that may hold additional information relevant to the request. To successfully complete the search, the client must be able to follow the continuation reference.

A referral is returned when a server is contacted by an LDAP client using a base object that is not contained by the server. Search references are a mechanism for partitioning the directory information tree (DIT) among multiple servers, and allow a search that has started on one server to be continued over one or more platforms.

Domino Directory services support providing referrals in response to queries. Directory Assistance is the mechanism used to define the referrals provided to the client. Domino Directory services in Release 5.x do not provide continuation references. The ability to follow continuation references was added to the Notes client in Release 5.0.5.

---

## 2.5 Directory searches

Within the rich directory services environment provided by Domino, it is useful to understand the order in which directories are searched and the various behaviors that are followed when addresses are resolved and identities are authenticated. While there is a consistent overall pattern, the specific behaviors differ depending on what components are present and how various parameters are set. This section attempts to outline the general pattern, what parameters affect that pattern, and to provide some specific information on mail address resolution and authentication.

Before that discussion begins there is one area that must be covered that may help clarify a number of user issues. This is the difference between the type ahead/type down functionality and the name resolution process invoked by pressing F9 or initiating a mail send.

The first point is that these are two distinct and separate functions with different design points and different behavior patterns. They are not intended to work the same and they do not provide identical results.

Type ahead/type down is a client-based function designed to provide a user addressing a message with a level of assistance in rapidly addressing a message. It is more limited in capability than “F9” or mail send. For example,

type ahead does not search LDAP directories. If a user Directory Catalog is configured on the client, type ahead will not extend its search to the server when looking for names or providing a list of potentially ambiguous names.

“F9” is designed to invoke the same NameLookUp code and name resolution processes that will be used by the router when attempting to transmit the message. It is a more robust and extensive functionality designed to eliminate mail address resolution issues to the maximum extent possible. It will search LDAP directories identified in Directory Assistance and will search both client and server based directories. It will generally return a longer list of potentially ambiguous names than type ahead, from a wider range of sources. The rest of this section will be focused primarily on how this more extensive process functions.

Users should be made aware of the differences and should be educated that while type ahead provides an excellent tool to assist in addressing messages, if there is any question in their minds, they should press “F9” to invoke a more complete search of all directory sources.

### **2.5.1 Directory search order**

The order in which directories are searched, especially when you are using a combination of Domino directories, Directory Catalogs and Directory Assistance, can be difficult to sort out. Domino uses slightly different procedures depending upon the combination of directories you are using.

At a high level, the search sequence is always the same regardless of whether the search is initiated by a Notes client, a Web client or other LDAP processes. These general steps are outlined in the following list. The first two steps apply specifically to Notes clients, while the last three apply to all clients. If any of the directory components are not present, that step in the lookup sequence will just drop away. Later in this section two specific cases are outlined in more detail.

1. The personal address book on the client machine
2. The user Directory Catalog on the client machine, if it is included in the notes.ini file in the names statement
3. The primary Domino Directory on the user's mail server or on the directory server, if that has been configured
4. The Directory Catalog on the user's mail server or on the directory server, if that has been configured
5. Directories listed in the server's Directory Assistance database that are not in the Directory Catalog

If the name entered in the mail addressing is a common name, not a hierarchical name, all directories in the Directory Assistance database will be searched according to their search order.

If the name entered is hierarchical, only the directories with rules that explicitly match the name, from most specific to least specific, will be searched. If two rules are equally specific, they will be searched in the search order specified in Directory Assistance.

## 2.5.2 Settings affecting the search sequence

There are several settings that impact the search sequence used by “F9” and the mail send function, as well as type ahead. These settings can be found on the location document, on the User Preferences dialogue and on the Configuration settings document for the server. It is worthwhile to take a minute to review them

On the Mail tab of the Location document there are two settings that impact type ahead and one that impacts the name lookup function invoked by “F9” or mail send.

*Recipient name type ahead:* This setting provides three choices: “Disabled”, “Local Only”, and “Local then Server”. The first is self explanatory. “Local Only” enables type ahead to search any address book listed in the “Local address book” field on the “Mail and News” panel of the “User Preferences” dialogue. This list can include the personal address book and one or more Directory Catalogs. The Directory Catalogs are searched in the order entered in the field. “Local then Server” enables type ahead to extend its search to directories, Directory Catalogs and secondary Domino directories configured in Directory Assistance that are located on the Domino server. By default the server will be the user’s mail server unless a directory server is specified on the “Servers” tab of the location document.

**Note:** If the user configures a local copy of a Directory Catalog, then the type ahead functionality will ignore the “Local then Server” setting and restrict its operation to the client.

*Activate recipient name type ahead:* This setting provides two choices: “On Each Character” and “On Delimiter”. If “On Each Character” is selected, then type ahead will attempt to find a match as each character is entered (there is a minimal delay before the search is started allowing additional characters to be entered). Type ahead will present the first match it finds in the first directory it finds that match, and will continue to refine those matches as each character is entered. If a Directory Catalog is configured, type ahead will match the character sequence to the sort

order for that Directory Catalog. For example, if the characters entered are “Jo” and the Directory Catalog is sorted by Last Name, it might present “Jones, Jim”, while if the Directory Catalog were sorted by Full Name it might present “Joseph Smith”.

There are two things to keep in mind here. First, if type ahead completes a name and you then enter a delimiter (you type Keith, it returns Keith Smith and you press comma) it will only search for ambiguous names that match Keith Smith -- it does not present all the Keiths, even though that is all you physically entered. Second, if you get into the “family” of names, it is frequently easier to use “type down” to make the actual selection. “Type down” allows you to use the up and down arrow keys to scroll through the names in the selected directory that are just before and just after the one displayed.

If “On Delimiter” is selected, type ahead will not be activated until a delimiter such as a comma or carriage return is entered. This gives the user more control over the point at which name resolution is initiated. Also, the presented ambiguous name dialog will contain any name that is ambiguous based on the entry made to that point.

*Recipient name lookup:* This setting impacts the behavior of the name lookup function that is invoked when “F9” or mail send is used. There are two choices here, “Stop after first match” and “Exhaustively search all address books”. “Stop after first match” will cause the name lookup function to restrict its name resolution activities to be restricted to the directory in which it finds the first match for the value entered. For example, if the user enters “John Smith” and presses “F9”, and the first John Smith is found in the primary domain directory, then any further operations, such as searching for ambiguous names, will only be applied to that primary domain directory. John Smiths that may occur in the Directory Catalog or in secondary directories accessed via Directory Assistance will be ignored. If “Exhaustively search all address books” is selected, then all accessible address books will be searched during name lookup operations.

**Note:** Type ahead is hard coded to follow the same behavior as “Stop after first match”.

On the “Mail and News” panel of the User Preferences dialogue, the user can specify which local address books will be used by type ahead by including them in the “Local address books” field. As noted above, if the user configures a local copy of a Directory Catalog, then the type ahead functionality will ignore the “Local then Server” setting and restrict its operation to the client.

In the Configuration Settings document for the server, the administration can impact the behavior of the router during its name resolution process. In steps 3 through 6 listed previously, the router, by default, has the option “Exhaustive lookup” disabled. This setting can be enabled, but it has a definite performance impact on the router. You can enable it via the Router/SMTP basics tab on a configuration settings document, as seen in Figure 1.

The screenshot shows the 'CONFIGURATION SETTINGS: ITS' document with the 'Router/SMTP Basics' tab selected. The settings are as follows:

Router/SMTP Basics	
Number of mailboxes:	<input type="text" value="1"/>
SMTP used when sending messages outside of the local internet domain:	<input checked="" type="checkbox"/> Enabled
SMTP allowed within the local internet domain:	<input type="checkbox"/> Disabled
Servers within the local Notes domain are reachable via SMTP over TCP/IP:	<input checked="" type="checkbox"/> Always
Address lookup:	<input checked="" type="checkbox"/> Fullname then Local Part
Exhaustive lookup:	<input checked="" type="checkbox"/> Enabled
Relay host for messages leaving the local internet domain:	<input type="text" value=""/>
Local Internet domain smart host:	<input type="text" value=""/>
Smart host is used for all local internet domain recipients:	<input type="checkbox"/> Disabled
Host name lookup:	<input checked="" type="checkbox"/> Dynamic then local

Figure 1. Enabling exhaustive lookup

**Note:** When the router uses the Directory Catalog, it performs an exhaustive lookup of all entries in the Directory Catalog, even if “Exhaustive lookup” is disabled. This is used as a default setting because the router can do the exhaustive lookup quickly in one database.

### 2.5.3 Search sequence for address verification in an LDAP directory

If a mail item is addressed to a user that exists in an LDAP directory, Directory Assistance accesses the LDAP directory to verify the addresses. Address verification occurs only when the Notes user presses F9 or before the Notes client sends the mail. Domino does not use type ahead addressing to resolve the addresses of users in LDAP directories.

A server running the LDAP service searches directories in the same general order described earlier. However, there are some nuances at each step that it

could be helpful to understand. The following describes the process in more detail:

1. The primary Domino Directory on the server
2. Directory Catalog on the server  
If the LDAP user searches for an attribute that maps to a field that is not configured in the Directory Catalog, and a secondary Domino Directory is configured in the Directory Assistance database as well as in the Directory Catalog, the search continues to the complete entries in the secondary Domino Directory itself.
3. Domino directories defined in the server's Directory Assistance database that are not included in the Directory Catalog  
If an LDAP user specifies a search base, only Domino directories with assigned naming rules that correspond to the search base are searched.
4. If the search is not successful in any Domino Directory, the server can pick an LDAP directory enabled for LDAP clients in the Directory Assistance database to refer clients to and the clients can then connect to the directory server themselves.
  - If an LDAP user specifies a search base, the server picks an LDAP directory enabled for LDAP users with a naming rule that matches the specified search base.
  - If there is no such directory, the server doesn't return a referral.

#### **2.5.4 Search sequence for authentication for a Web client**

As with address verification in an LDAP directory, the server follows the same general search order when a Web user gets authenticated by a Domino Web server. The following list outlines the specifics of this process:

1. The primary Domino Directory on the server
2. Directory Catalog on the server.
  - If the server finds the name, it refers to the Directory Assistance database to determine if the Domino Directory from which the name came is configured with a naming rule trusted for authentication that matches the user name.
  - If the name is found to be trusted, the server looks up the HTTP password or x.509 certificate in the secondary Domino Directory that the Directory Catalog entry has been derived from.
  - If an HTTP password is stored in the Directory Catalog itself, the server looks up the password in the Directory Catalog rather than in the

Domino Directory. But it will not authenticate the client if it does not find a trusted matching naming rule for the Domino Directory in the Directory Assistance database.

3. All other directories defined in the Directory Assistance that are not included in the Directory Catalog and that have a naming rule that is trusted for authentication that matches the Web user name.

If there is more than one directory assigned in a trusted naming rule that matches the username, the directory with the most specific matching rule is searched first.

Table 1 has a comparison between the functionality of the Directory Catalogs and Directory Assistance

*Table 1. Directory Catalog vs. Directory Assistance*

<b>Task</b>	<b>User Directory Catalog</b>	<b>Server Directory Catalog</b>	<b>Directory Assistance</b>
Find entries in secondary Domino directories on behalf of Notes users for mail addressing.	Yes	Yes	Yes
Find entries in LDAP directories on behalf of Notes users for mail addressing.	No	No	Yes
Find entries in secondary Domino directories on behalf of LDAP clients.	No	Yes	Yes
Refer LDAP clients to LDAP directories.	No	No	Yes
Authenticate Web clients using entries from secondary Domino directories.	No	Yes *	Yes
Authenticate Web clients using entries from LDAP directories.	No	No	Yes
“Recipient name type ahead” addressing.	Yes	Yes **	Yes

\*Also requires Directory Assistance to define the names that can be authenticated.

\*\* Only for Notes users who don’t use the mobile directory catalog.



### 2.5.5 Search sequence for group authentication

While the term “group authentication” is generally used, it is actually a misnomer. Individual identities are authenticated, and then a list of the groups an authenticated identity belongs to is created. The individual authenticated identity and the list of associated groups is then used to authorize access to a database (more on this later). The process of opening up group lists to determine if a particular authenticated identity belongs to a group is commonly called “expanding the group”, or simply “group expansion”.

Another characteristic of groups is that they can contain other groups. This is known as nesting groups, where one group resides inside another. Some applications, like Domino, allow expansion of nested groups as well as simple groups.

Domino uses a different and more restrictive pattern to locate groups to which an authenticated identity belongs for purposes of authorizing access to applications and databases. For a Notes user, once an individual identity has been authenticated, Domino will check groups and nested groups in the primary Domino Directory. For a web user, once an individual identity has been authenticated Domino will check groups and nested groups in the primary Domino Directory and in one LDAP directory that is accessible via Directory Assistance. The specific LDAP directory is determined by settings on the Directory Assistance document. Directory Assistance documents for the LDAP protocol have a selection on the “Basics” tab for group expansion. If “Yes” is entered, then a second selection for “Nest group expansion” appears.

The Directory Assistance database validation logic precludes the administrator from designating more than one directory for group expansion.

---

## 2.6 Authentication and security services

The default Notes client-to-server or server-to-server authentication procedure depends on a certificate, an electronic stamp that indicates a trust relationship between the two entities. The certificate is stored in the Notes ID file. Notes makes use of a number of cryptographic techniques, such as public key and symmetric key encryption, digital signatures, and public key certificates.

A Notes certificate contains the following:

- The certificate owner's name and details
- The certificate owner's public key
- The certifier's name and details
- The certifier's public key

- The certificate expiration date

It is possible to switch off certificate-based authentication for Notes users, which will enable Notes users to remain anonymous until the user attempts to perform an operation that is not allowed to be performed by anonymous users. This can be done by going into the security settings section of the server document and changing the “Allow anonymous Notes connections” field value to “Yes.” By default this is set to “No.”

Security Settings	
Compare Notes public keys against those stored in Directory:	<input type="radio"/> Yes <input checked="" type="radio"/> No
Allow anonymous Notes connections:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Check passwords on Notes IDs:	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled

For non-Notes clients, there are three levels of security and authentication:

- Anonymous access, which is normally allowed on servers available to the general public. Access remains anonymous until the client attempts to perform an operation that is not allowed to be performed anonymously. At the database level, access is granted to an anonymous user or server by entering the name Anonymous in the access control list. If there is no such entry, anonymous users are granted the privileges of the Default entry. The Domino server uses the name “Anonymous” solely for access control checks. The HTTP, POP3, IMAP, LDAP and NNTP clients use anonymous as their default access method.

You cannot use anonymous client authentication for SMTP and IIOP connections.

- Name and password authentication happens when a server is configured to not allow anonymous access or when an anonymous user attempts to perform an operation that is not allowed to be performed anonymously. As soon as a non-Notes client tries to access the server, the user is challenged to enter his name and password. The server then compares this combination of username and password with the entries in the primary Domino Directory or, for Web clients only, directories that are configured in Directory Assistance. The server can use any combination of primary and secondary Domino directories, Directory Catalogs and LDAP directories as long as the authentication rules for these directories are defined within the Directory Assistance database.

- Certificate-based authentication for non-Notes clients uses the secure sockets layer. Both the client and the server need to have a common certificate from a trusted third party. The third party vendors in this case are referred to as the certificate authority (CA). You can use a third-party, commercial company such as VeriSign as the external certificate authority, or you can use a Domino certificate authority as an internal certificate authority in your organization.

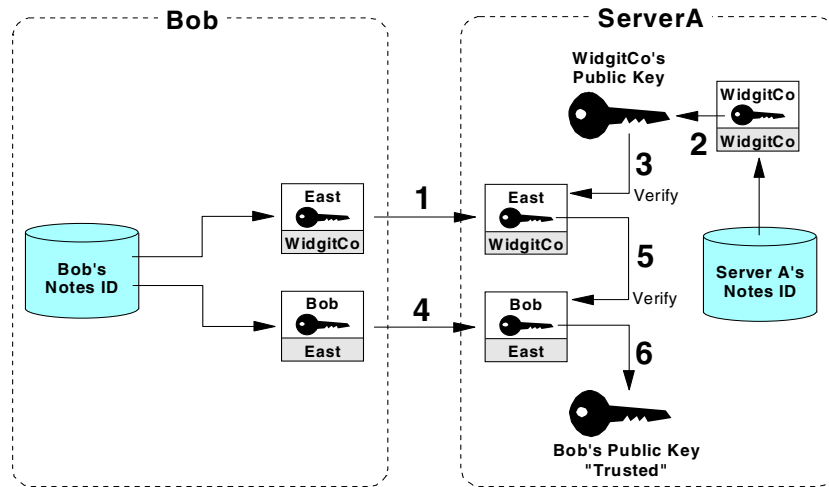
### **2.6.1 Authentication for applications via NRPC**

When a Notes user attempts to access an application via Notes remote procedure call (NRPC), the client and server present their certificates to each other. By examining certificates, the client and server will identify and authenticate each other.

There are two phases in verifying a user or a server's identity in Notes. The first phase, called validation, is the process of reliably determining the sender's public key. In other words, the validation is the preparation phase for the actual authentication. Notes uses the following rules when deciding to trust a public key:

1. Trust the public key of any of your ancestors in the hierarchical name tree because they are stored in your ID file.
2. Trust any public key obtained from a valid certificate issued by any of your ancestors in the hierarchical name tree.
3. Trust any public key certified by any trusted certifier and belonging to one of the certifier's descendants.

Here is an example of how these rules are applied in the validation process. The user ID file for Bob Smith contains everything he needs to identify himself and establish his credentials. When he requests a session with a server, the first step is to send to the server all of the certificates from the ID file (both the user's own certificate and the chain of certifiers' certificates that support it). The figure below illustrates the validation process that follows.



The numbered steps in the figure are described as follows:

1. ServerA reads the East certificate that Bob Smith sent from its ID file. This was signed by WidgitCo. ServerA is interested in it because East is the certifier of Bob's certificate.
2. ServerA reads the WidgitCo public key from its own ID file. ServerA will trust the public key of any ancestor that is stored in its ID file.
3. ServerA uses the public key of WidgitCo (which is trusted because it is in the server's ID file) to verify that the certificate of East/WidgitCo is valid. If you trust the public key of the ancestor, you will trust any public key obtained from certificates issued by the ancestor.
4. ServerA reads the certificate that was sent from Bob Smith's ID file. This was signed by East.
5. ServerA uses the public key of East/WidgitCo, which now is trusted, to verify that the Bob Smith/East/WidgitCo certificate is valid. Trust any public key certified by any trusted certifier and belonging to one of the certifier's descendants.

ServerA has now reliably learned Bob Smith's public key. The same process is followed in reverse so that Bob can reliably learn ServerA's public key.

The second phase in verifying a user or a server's identity in Notes is authentication. Authentication is proof of identity. The validation process described above has not completely proved who each of the session partners is, because all they have presented so far is certificates. A certificate associates the user with a public key and tells the recipient that the public key can be trusted, but in order to prove that users really are who they claim to be

they must show that they hold the private key that matches the public key in the certificate. The authentication process achieves this with a challenge and response dialog between a workstation and a server, or between two servers when either is running database replication or mail routing.

To continue the previous example of Bob Smith accessing ServerA, see Figure 2. The following is a simplification of the actual process, and is intended to illustrate what happens in a manner that's easy to understand.

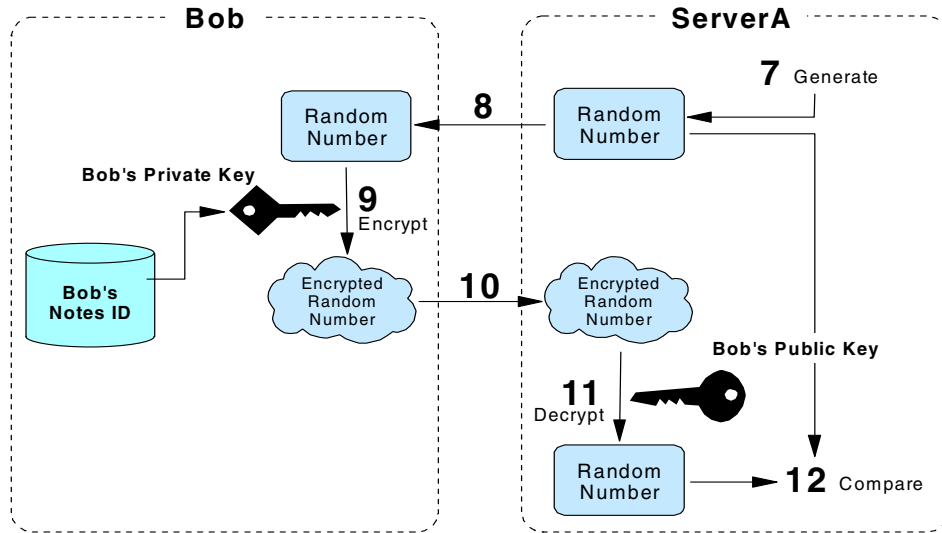


Figure 2. Authentication process flow

1. ServerA generates a random number and a session key, and encrypts both with Bob's public key.
2. ServerA sends the encrypted random number to Bob Smith.
3. Bob receives the challenge and decrypts it with his private key.
4. Bob Smith sends back the decrypted number to ServerA.
5. ServerA compares Bob's response to the original random number.
6. If the result is the same as the original random number, ServerA can trust that Bob Smith really is who he claims to be.

As with validation, authentication is a two-way procedure. Bob now authenticates ServerA using the same challenge/response process in reverse.

The actual algorithm is complex but efficient. It avoids any RSA operations on subsequent authentications between the same client-server pair. It also establishes a session key that can be used to optimally encrypt the messages that follow authentication.

If authentication was successful, you now have access to the server. The next level of security is authorization. When a client or server attempts to access an application or perform an action in it, the server verifies that the user or server request is allowed. A database access control list determines the level of access that users, groups, and servers have. Someone with manager access to the database assigns access levels to the users, groups, and servers listed in the access control list.

There are seven main levels of access that a database administrator can assign to a person, server, or group. They are described in Table 2.

*Table 2. Levels of user access to a database*

<b>Level</b>	<b>User actions allowed</b>	<b>Server actions allowed</b>
No Access	Not access the database at all.	Not access the replica at all.
Depositor	Create documents but cannot read, edit, or delete documents, including those they create.	Not receive changes.
Reader	Read documents, but cannot create, edit, or delete them.	Pull changes from the replica but not send changes to it.
Author	Create and read documents, but can only edit their own documents if they are listed in an Authors field on that document.	Replicate new documents.
Editor	Create, read, and edit all documents unless there are restrictions on specific documents.	Replicate all new and changed documents.
Designer	Have Editor access to documents, except where restrictions exist for specific documents, and they can modify the database design, but they cannot delete the database or modify the ACL.	Replicate design changes as well as all new and changed documents, but not ACL changes.
Manager	Perform all operations on the database, including modifying ACLs and deleting the database.	Replicate all changes to the database and the ACL.

To further improve the security in an application, you can also make use of the “User type” and “Access” and “Roles” in the access control list. For detailed information on the levels of access within a Domino Directory or application, refer to the following redbooks: *Lotus Notes and Domino R5 Security Infrastructure Revealed*, SG24-5341, and *Lotus Domino R5.0: A Developer’s Handbook*, SG24-5331.

### 2.6.2 Authentication for LDAP clients

Name and password authentication between a Domino server and any LDAP client is a very simple process. The client submits a name and password to the server, which checks the primary directory for a record with the same name and password.

To understand the differences and dependencies of each of the Internet client types, you have to understand the differences between the different types of firewalls that can be used.

The first type of firewall is a packet-filtering firewall, which examines the destination and content of each network packet transmitted over the network. It then checks whether the network allows delivery to that destination and allows that type of information to enter or exit your network. If the packet passes these tests, the packet-filtering firewall allows the packet to proceed to the destination. You typically use network router software to implement a packet-filtering firewall.

The second type of firewall is an application proxy firewall. This is a server program that understands the type of information that you are transmitting, for example, NRPC or HTTP format information, and controls the flow of information between internal and external clients and servers. An application proxy works the same as a packet filtering firewall, except the application proxy delivers the packet to the destination.

- You can set up a Domino passthru server as an application proxy for Notes NRPCs.
- You can use an HTTP proxy to communicate using:
  - HTTP secured with SSL
  - Internet protocols secured with SSL
  - Notes remote procedure calls (NRPC)

The third type of firewall is a circuit-level proxy. It is similar to an application proxy, except that it does not need to understand the type of information that is being transmitted. For example, a socks server can act as a circuit-level proxy.

- You can use a circuit-level proxy to communicate using Internet protocols with TCP/IP, which includes IMAP, LDAP, POP3, NNTP, SMTP, IIOP and HTTP.
- You can also use a circuit-level proxy for Notes NRPCs or with the Internet protocols secured with SSL.

The following Internet protocols are supported by Domino and SSL:

- Web server and Web navigator (HTTP)
- Internet inter-ORB protocol (IIOP)
- Internet message access protocol (IMAP)
- Lightweight directory access protocol (LDAP)
- Network news transfer protocol (NNTP)
- Post office protocol 3 (POP3)
- Simple mail transport protocol (SMTP)
- Simple authentication and security layer (SASL)

Domino uses SASL automatically if SSL with client authentication is set up on the server and if the LDAP client supports the protocol.

Table 3 shows the types of access that the servers will allow for the different client types.

*Table 3. Types of access.*

Client protocol	Server allows anonymous client access	Server can use name and password authentication	Server can use SSL authentication
<b>HTTP</b>	Yes	Yes	Yes
<b>LDAP</b>	Yes	Yes	Yes
<b>NNTP</b>	Yes	Yes	Yes
<b>IIOP</b>	Yes	Yes	No
<b>SMTP</b>	Yes	Yes	No
<b>POP3</b>	No	Yes	Yes
<b>IMAP</b>	No	Yes	Yes

- You cannot use application proxies for IMAP, LDAP, POP3 and NNTP unless the protocol is set up to use SSL.



- You can use packet filtering or circuit level proxies for IMAP, LDAP, POP3 and NNTP without SSL.
- Name and password authentication is not supported when a Domino server acts as an SMTP client, which is what happens when a Domino server connects to an SMTP server to route mail.
- Name-and-password security is only supported when a Domino server acts as an SMTP server, which is when SMTP clients access a Domino server.

For SSL client authentication, the Notes or Internet client must have the following:

- An Internet certificate issued from a Domino or third party certificate authority
- A trusted root certificate for a Domino or third party certificate authority (Notes clients only)
- A cross-certificate for the Domino or third party certificate authority created from the trusted root certificate

The trusted root certificate is not necessary for Notes clients after you create the cross-certificate.

- Software, such as a Web browser or a Notes workstation, that supports the use of SSL

### **2.6.3 Authentication in the Domino Directory services**

When authenticating a person, a server always searches the primary Domino Directory, then the Directory Catalog, before it uses Directory Assistance. The server Directory Catalog is designed so that a server needs to search only the Directory Catalog and not multiple secondary directories. Each entry in the Directory Catalog includes the replica ID of the Domino Directory from which the entry was derived and the unique ID associated with a replicated document.

Figure 3 on page 30 shows a graphical representation of the lookup sequence in the authentication process.

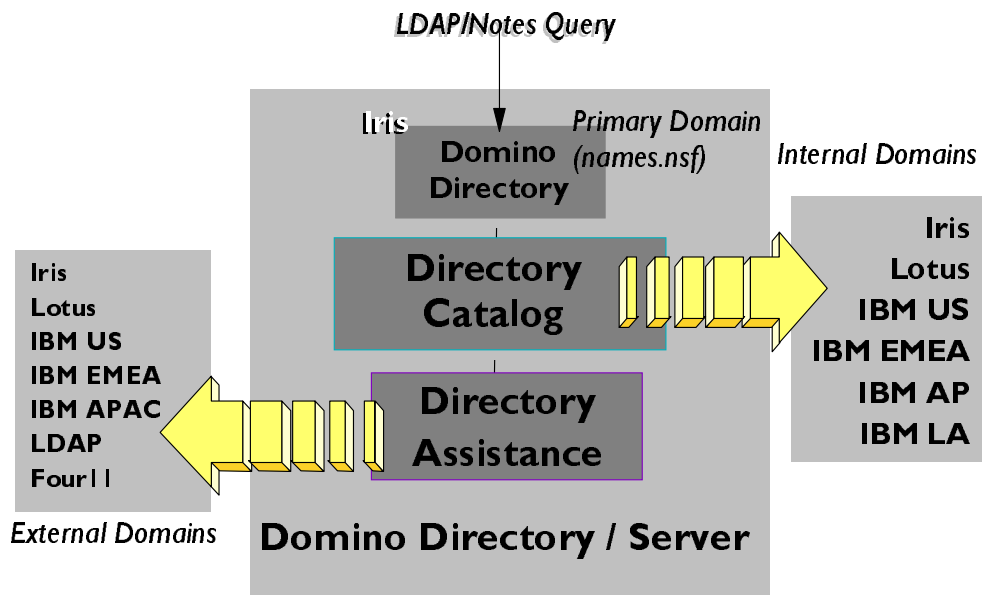


Figure 3. Lookup sequence in the authentication process

As an example, the IBM environment consists of multiple domains, with a common Directory Catalog that includes all their Domino directories. The Directory Catalog has been configured with the default settings. Using HTTP, an IBM user in France attempts to access an application in Germany. The primary Domino Directory will not be able to verify the username and password. This is because the server is outside the user's home domain. The Directory Catalog will be searched and the user name will be found, but not his HTTP password. When the Directory Catalog does not have all the information to satisfy a search, the server uses the document information to very quickly access the complete entry in the secondary Domino Directory. This is possible because the replica ID of the Domino Directory from which the entry was derived and the unique ID associated with a replicated document is kept in each entry in the Directory Catalog.

## 2.7 Domino Directory as an enterprise directory

As an enterprise directory, the R5 Domino Directory is a secure, standards-based directory that is scalable, extensible, and distributed.

In a heterogeneous networking environment with other directory systems, the Domino Directory can be used to serve as the integration point for directory

synchronization, administration and authentication. Some of the key features that position the Domino Directory as an enterprise directory are:

- Support for X.500 naming conventions
- LDAP protocol support in both the client and the server, providing lookup support for non-Notes clients and servers
- Rule-based domain relationships for faster lookups across large namespaces
- Hierarchical naming and trust between domains to support the relationship of entries across domains
- Full support for a public key infrastructure for robust security
- An extensible directory schema ideal for customizing the directory
- Multi master replication, a key element in reliable directory synchronization

The directory utilizes the extensive Domino access control, which allows multiple levels of granularity. Authentication can be as simple as allowing anonymous access or it can be as secure as certificate-based authentication with SSL.

The Domino Directory is compliant with LDAP version 3 standards. As such, it supports LDAP clients such as Netscape, MSFT and Eudora. The directory also interacts with other LDAP-compliant servers. This provides user and group authentication services for LDAP-compliant software. For examples, see 7.2, “Integrating with other directories and applications” on page 110.

The Domino Directory is distributed by Domino replication. It can also act as a single repository for all user and group definitions. The directory can store X.509 certificates from Domino and other certificate authorities. The directory schema may be extended to define new objects or additional attributes for existing objects.

Administration of the Domino Directory can be done using the Domino administration client, the Domino Web administration client, or other LDAP administration clients.



---

## Chapter 3. Planning your directory services

When planning your directory services, there is much more to it than just an information technology solution. Each deployment might be different and is very dependent on the organizational culture and usage patterns of the information technology infrastructure.

This chapter cannot provide you with a perfect set of questions and answers for your organization, but it can provide you with a set of questions and a list of considerations that you can customize to suit your environment. In smaller organizations, a lot of these questions will have one-line answers, but as components are added to a configuration, the complexity increases which increases the complexity of the answers.

---

### 3.1 What are the requirements and business drivers?

The requirements and business drivers must be identified before you even start to plan the technical solution. The Domino Directory services solution can only be successful if it supports the business needs and if it is in turn supported by the business.

You can only find the real business drivers and requirements if you discuss them with a cross-section of the organization—groups like the end users, managers, project personnel, and human resources. This is the only way to identify the business requirements and the technology requirements and to understand the existing technology base.

The questions you need to ask during this phase are:

- Who is the sponsor of the solution? Is it a technical department, management, or a user group driven by a common need?
- Who will be paying for the solution?
- Does the sponsor have a realistic understanding of Domino Directory services capabilities and limitations?
- What are the goals of the sponsor and the organization?
- Try to understand the suggested solution deployment time frame. Is it a workable time frame?
- Is this a strategic/long term solution or a tactical/short term solution?
- Is there an application driving the need for the Directory Assistance services?
- Is training needed for the administrators of the Domino Directory services?
- Has the normal user population been consulted in these changes and do we have an understanding of what they need?

- What are the organizational dynamics which need to be addressed? Will all the groups work together or is this an issue that needs to be addressed?

For example, the information technology company “F.R.E. Merchants” wants to use their Domino Directory services solution as an enterprise directory. To succeed, they will need buy-in from:

- Management, in the form of a sponsor and an entity that will pay for the solution.
- The human resources department, network administration group, mainframe administration group and the notes administration group, as co-owners of the data in the directory solution.
- The user population, as users of the data in the directory solution.

This would be needed because an enterprise directory implies a single point of information management and administration, and all these groups will have to agree on roles and responsibilities for the solution to be successful.

---

### 3.2 Discovering your organization

Now that you have an understanding of the political and human factors within your organization, it is time to look at your organization’s current information technology environment:

- Why do I need to identify my existing environment? This is to ensure that the new solution will fit into the current environment.
- Why change from the existing environment? Changes in the existing environment will only be needed if anything has to be added or upgraded to allow the new solution to be effective.
- What are the interoperability issues within your company? What are the interoperability issues between your company and external business partners?

You need to identify current and additional resources needed, such as:

- Additional network hardware and software requirements
  - Does the current network have enough capacity for the new solution?
  - Are you using the best network operating system for your solution?
  - How do mobile users connect to the network?
- Network needs, with a clear picture of your LAN and WAN topologies, including:

- Domino servers
- Mainframes
- File and print servers
- Bridges and routers
- Leased lines and phone lines
- Workstations
- The type of network cabling that you are using, for example, Ethernet 10baseT or Token ring, and so forth
- The current network usage within each LAN
- The type of protocols used within the network
- The location of sites connected in the WAN
- The WAN bandwidth availability and network usage between sites
- The split between protocols in the WAN usage between sites
- The number of administrative resources used in the current network topology
- What network management and monitoring tools are in place
- Additional Domino server requirements.
  - Does the current setup have enough capacity for the new solution?
  - Do we need to add additional Domino servers?
  - What operating system are you using at the moment?
  - What operating system will suit your needs the best?
  - How many processors does the Domino server have?
  - How much RAM and disk capacity does the Domino server have?
  - What are the current Domino servers' response times and workload?
  - How do mobile users connect to the Domino servers?
- How many people or groups within the organization are involved in maintaining the current solution?
  - Does the sponsor 'own' all these groups?
  - What is the skills base that will support the changes?
  - Do you need to train any users, network administrators or other people?
- User workstation hardware and software requirements.
  - How many of the users can continue on the current hardware platform?
  - How much RAM, disk capacity and processor speed is needed?
  - How many of the users are on the correct software code levels?
  - What operating system software is needed?
- What are the current constraints associated with the WAN, LAN, remote access, and the Domino services?
- What is the current backup and archiving strategy?

There are multiple types of directories:

- System directories - which is typically what you have on every operating system and network protocol where users and groups are defined for access to the resources. These directories are normally focused on security. They also include profile information, which is used to initialize a user session, set up devices and network drives, and so forth.
- Phone book directories - These directories are used for lookup purposes. An example is a phone directory that is split into yellow and white pages. The white pages refer to user details, sorted by surname, and the yellow pages is sorted by category.
- User directories - Mail systems use an address book, for example. Notes and most of the other e-mail systems have, in one form or another, a personal address book in which you store people's details and e-mail addresses.
- Application directories - usually contain information about users, groups and peripherals that is used by the application. Each application directory can be completely unique from the next one. For example, when you use a specific printer on the network, how does the application remember which printer you used? This information and all your default settings are stored in a directory within the application.

Identify your current directories:

- Where in the organization do you find the existing directories?
  - Domino user administration group
  - Human resources department
  - Server administrators - you might have one in each department
  - Network operating group
  - Mainframe administrators
  - Mainframe application administrators - each mainframe application might have a different set of administrators.
  - Organizational directory group - you might have a central organizational directory for people's organizational and contact details
  - Every operating system has a directory, so this may need to be addressed as well.
- Identify the sources of your directories:
  - An e-mail or messaging system
  - A human resources or payroll system
  - A network operating system
  - A domain name service
  - An application or multiple applications



- Which directory should be considered as the organization's source directory?
- How and where is the directory information stored?
  - How accessible are the directories?
  - Can they be more accessible?
  - Are the directories secure enough?
  - How many domains, environments or directories are currently managed?

---

### 3.3 Defining directory ownership and administration

Define who owns each part of the data:

- What data is critical to business operations?
- Who is currently maintaining the business-critical data?
- Is current administration of data limited to a particular group or department or functional area?
- What information do you need to make available to people?
- What information do you need to restrict?
- Which groups or departments can see what part of the data?
- What data is private? Keep in mind that different countries and states have different privacy laws.
- Who has to make the final decision about data privacy?
- What guarantees are there for data currency at the moment?
- Who is going to manage the end user expectations?
- Is it possible to link or connect the existing directories and the new solution?
- Do you have the expertise to manage data integration?
- If you are considering migrating directories, what security considerations are there?
- Are there any other security implications and considerations?
- Should you allow remote access to the directories?
- What is the current user interface used to manage the directories?
- Can all the directories be linked into a Domino Directory services solution?

Define who will administer the data:

- Who needs update access to the data?
  - Is it a single group?
  - Are there multiple groups that need access to all the data?
  - Are there specific groups that need access to specific parts of the data?

- How many directories are you going to administer centrally?
  - Some directories might not be part of the general directory services.
- Will the same group or department that is responsible for updates to existing entries also be responsible for new additions to the directory?
- Which groups or departments should be able to see what part of the data?
- Should we allow remote administration access to the directories?
- Are you going to use multiple groups to manage data accuracy?
- Do you need to define different levels of access to specific parts of the directory to add granularity to the directory information security?
- What information will be entered in what directories?
- How can you limit the number of people entering data?
- What is the end user expectation regarding the data being up to date?

---

### 3.4 Directory aggregation, synchronization and publication

How do you populate and build your directories?

- What underlying architecture exists in the current directories?
- Which directory entries are common across directories?
- Is there a unique key to logically link these entries?
- Does the unique key vary between directories?
  - Is the same key that is a unique link between directories A and B valid when comparing directories A and C?
- What methods are currently being used to aggregate directories?
- What resources are currently being used in the aggregation?
- Can this administration be centralized?
- What tools can be used to aggregate directories?
- What resources will be needed?
- How many different directories will be aggregated?
- What will each directory be used for?
- What will the audience be for each directory?
- What underlying architecture will be used for directory aggregation?
- How long will each initial aggregation take?
- How often do you need to aggregate your directories?
- How long will each subsequent update take?

How do you keep your directories synchronized?

- What methods are currently being used to synchronize directories?
- How frequently are directories being synchronized?
- How many different directories need to be synchronized?
- On how many platforms are the directories that need to be synchronized?
- How many resources are needed and how much time is spent on fixing synchronization problems?

- What is the end user experience on the data being up to date?
- Can the administration and synchronization be centralized?
- What tools can be used to synchronize different directories?
- How long will it take before all directories are synchronized?
- How often do you need to synchronize your directories?
- How many source directories are you going to use to synchronize across your organization?

What considerations are there when publishing your directories?

- Are there any current solutions in place?
- How frequently are directories currently published?
- How many of the existing directories are being published?
- Where are the directories being published?
  - Are they published on paper?
  - Are they published on the intranet?
  - Are they published on the internet?
- What information is disclosed in the publications?
- Has the information disclosed in publications been approved by the organization's security group?
- Has the information disclosed in publications been approved by the organization's legal group?
- What resources are currently being used in the publication process?
- Is the publication process centralized or are there many parts of the organization doing their own thing?
- What are the implications if the publication process is centralized?
- What tools are you currently using to publish directory information?
- How many different directories are published and how often are they published?

---

### 3.5 Planning your Domino Directory services

How you approach the planning of your Domino Directory services depends on three factors:

1. Is this a new deployment of a Domino infrastructure?
2. Are you enhancing or expanding your directory services?
3. Are you deploying Domino as an enterprise directory?

Once you have answered these questions, you will have to go on to the detailed planning of your directory solution.

### **3.5.1 Is this a new deployment of a Domino infrastructure?**

If this is a new deployment or a first-time setup of your Domino infrastructure, you will have to think about the following issues:

- Do you have to design the topology from scratch?
- Do you have to design your hierarchy?
- Will the current organizational structure suit as a Domino hierarchy?
- Which solution suits your environment?
  - A hub and spoke topology?
  - A mesh or peer-to-peer topology?
  - A ring topology?
- Are there any bandwidth considerations?
  - Do you have enough available capacity on your LAN?
  - Do you have enough available capacity on your WAN?
- How many Notes named networks do you need?
  - This affects mail routing since servers in the same Notes named network route mail immediately.
  - Servers in different Notes named networks only route mail on schedule as defined in connection records.
  - With multiple Notes named networks, the administrative workload increases because you need to schedule mail routing and replication.
- How many Notes domains do you need?
  - In a single domain, direct mail routing leads to fewer bottlenecks and easier troubleshooting.
  - Replication and mail routing is easy to manage in a single domain.
  - You have more control over mail routing if you have multiple domains.
  - Since each Domino Directory represents a domain, your directories will be smaller, which in turn can lead to improved performance.
  - With multiple domains you need centralized administration or your support cost will dramatically increase.

### **3.5.2 Are you enhancing or expanding your directory services?**

If you are expanding or enhancing your Domino infrastructure, you will have to think about the following issues:

- Can you use your current solution and build on it?
- Do you have to redesign your organizational structure?
- Are there any bandwidth considerations with the new solution?

- Do you have enough capacity on your LAN?
- Do you have enough capacity on your WAN?
- Will your current servers be able to handle the extra workload?
- Do you have to deploy directory servers to manage the workload for your directory services?
- How many servers do you need to ensure high availability for your directory services?

Once you have considered the capacity issues for your new solution, the next step is to consider the actual directory services solution.

### **3.5.3 Are you deploying Domino as an enterprise directory?**

If you are deploying Domino as an enterprise directory, you will have to think about the following issues:

- How many different directories and sources do we have?
- Which directory or source is considered the master?
- Which directories do you want to integrate?
  - Not all directory types can be integrated, but the more directories you integrate, the smaller the administration overhead becomes.
  - For more information on how to integrate different types of directories, see 7.2, “Integrating with other directories and applications” on page 110.
- Can you use your current solution and build on it?
- Which methods or tools are you going to use to integrate these different directories?
- How often do you need to run the aggregation of the enterprise directory?
- How often do you need to synchronize between the directories?
- Do you have to deploy directory servers to manage the workload for your directory services?
- To ensure consistency and prevent replication/save conflicts, can you centralize your administration?
- How many servers do you need to ensure high availability for your directory services?
- Are you going to use your primary Domino Directory as the enterprise directory?
- Are you going to use a secondary directory as your enterprise directory?

- Are you going to use an extended Domino Directory as your enterprise directory? If you are going to extend the directory design by customizing it for your organization, you have to consider the following implications:
  - Be aware that modifications to any form in the Domino Directory can lead to unexpected problems. You need to be very sure of what you are changing.
  - Changing and adding views in the Domino directories will add to the directory size as well as the workload that each server in the domain will have to handle.
  - If you are modifying views, be aware that the hidden views in the Domino Directory should not be touched at all. They only exist for use by the server.

### **3.5.4 The Domino Directory services details**

Now that you have determined the topology of your Domino setup, defined the capacity issues for your solution, and decided which directories to integrate into your enterprise directory, you need to plan the actual Domino Directory services details.

What combination of directories are you going to use in your Domino Directory services?

- Are you going to be using multiple secondary Domino directories?
- Will you use them for mail addressing?
- Will you use them for authentication?
- Do you have a server with all the secondary Domino directories on it?
- How often are these directories replicated or synchronized?
- How many Directory Catalogs are you going to use?
- Are you using the server Directory Catalog as part of your authentication solution?
- How many different Directory Assistance databases are you going to use in your organization?
- On which servers are you deploying the Directory Assistance database?
- How many LDAP directories are you going to use?
- Are you going to merge any LDAP directories into secondary Domino directories?
  - How often are you going to aggregate these directories?
  - How often are you going to synchronize these directories?

- What software do you need to allow you to do the synchronization?
- What software do you need to allow you to do the aggregation?

What considerations are there for the Domino Directory?

- How static will the data in your Domino Directory be?
  - In a geographically dispersed environment, where you can have multiple network considerations, data that changes often can have severe implications on replication times.
- If you have multiple domains, are you making each domain's primary Domino Directory available in the other domains as a secondary Domino Directory?
- Are you going to use the secondary Domino directories for mail addressing?
- Are you going to use the secondary Domino directories for authentication?
- Are you going to centrally manage all your Domino directories?
- How often do these Domino directories need to be replicated?
- How many groups need update access to these directories?
- Are you going to use role definitions in the access control lists of these directories to refine your security?

What considerations are there for the Directory Catalogs?

- How do you need to configure the Directory Catalogs?
- What packing density are you going to use when you aggregate your Directory Catalogs?
  - If you have 255 documents in a source directory that are aggregated into one document in the Directory Catalog, you are 255 times more likely to have to replicate that single document than in a traditional directory.
- What is the default corporate usage when you think about mail addressing and type-ahead name resolution?
- How many Directory Catalogs are you going to use in your directory services?
- Are you going to use a dedicated directory server to aggregate the Directory Catalogs or do you have a server with enough capacity to handle this workload on top of what it already does?
- How many servers are you going to use to replicate the Directory Catalogs to other servers?

- How many servers are you going to use to replicate the Directory Catalogs to end users?
- How are you going to deploy your Directory Catalogs to other servers?
- How are you going to deploy your Directory Catalogs to the end users?
- Will your current replication topology be able to manage the additional replication load?
- How are you going to manage the additional network load when users start to replicate the Directory Catalog?
- If you have WAN and LAN constraints, how are you going to distribute the user Directory Catalog in those locations where you are restricted?
- Are you going to use each mail server to handle the replication of the user Directory Catalog?
- How can I control the number of updates on the Directory Catalog and therefor limit the end user impact on a daily basis?
- How often am I going to aggregate or update the Directory Catalogs?
- What size can the user Directory Catalog be before it will be considered to be too big? Can your average workstation handle a Directory Catalog of that size?
- What size can the server Directory Catalog be before it will be considered to be too big? Does your servers have enough capacity to handle that size?
- What data are you going to include in the user Directory Catalog?
  - Is the data you plan to include really needed for mail addressing?
  - Is the data you plan to include needed for people to contact each other?
  - Is the information you are adding into the Directory Catalog relatively static? If the information is very changeable, it can effect the amount of user replication that will happen with the server as well as impact the workstation while the client software is updating the full text index on the local Directory Catalog.
- How often are you going to aggregate the user Directory Catalog?
  - More regular aggregation will minimize the impact on the workstation, but may have a negative effect on the server, LAN, and WAN while replicating these changes throughout the domain.
  - Less frequent aggregation will minimize the impact on the server, LAN, and WAN, but may have a negative effect on the workstation while replicating these changes and updating the full text index.



- Are you going to use the server Directory Catalog as part of your authentication topology?
  - If so, you will have to consider including the HTTP password and certificate fields in the Directory Catalog.
  - This will have a big effect on the size of the server Directory Catalog.
- What data are you going to include in the server Directory Catalogs?
  - Is the data you plan to include really needed for mail addressing?
  - Is the data you plan to include needed for authentication purposes?
    - If the data is not in one of the above categories, it should not be included in the server Directory Catalog.
  - Is the information you are adding into the Directory Catalog relatively static?
    - If the information is volatile, it can affect the amount of replication that will happen between servers.
    - If the information is volatile, it can also affect server performance while the server updates the Directory Catalog's full text index.
- How often are you going to aggregate the server Directory Catalog?
  - More regular aggregation will minimize the impact of the replication time between servers.
  - Less frequent aggregation will minimize the impact on the LAN and WAN, but may have a negative effect on the server while updating the full text index.
- Will the data included in your Directory Catalogs be relatively static?
  - If not, you need to think about the impact of replication on your servers and workstations, and on the LAN and WAN.

What are the considerations for Directory Catalog and the LDAP service?

- How many different Directory Assistance databases are you going to use?
- Do you need multiple replicas throughout the domain or do you need replicas on specific servers?
- Do you need multiple Directory Assistance databases with different rules on different servers?
- Which directories do you need to include in your Directory Assistance database and LDAP service?
- Design the LDAP schema to fit your environment.
  - How much information do you really need?

- How much of it will change regularly?
- Is there any real benefit if you include fields that are volatile?
- For more information on designing and maintaining an LDAP directory, see chapter 3 in the redbook *Understanding LDAP*, SG24-4986.
- In the Domino Directory services, which group should ultimately have the administrative responsibilities?
  - How many groups are you going to use to manage data accuracy in the primary Domino Directory?
  - How many groups are you going to use to manage data accuracy in secondary Domino directories?
  - How many groups are you going to use to manage data accuracy in the LDAP directories?
  - With synchronization, can you lower the number of groups that is needed for data maintenance between the different directories?
  - What tools and software are you going to use to synchronize data and centralize administration?

---

## Chapter 4. Setting up and configuring Domino Directory services

Domino Directory services require a Domino server on which to run. There are three varieties of Domino server available: Domino Mail Server, Domino Application Server, and Domino Enterprise Server. All three products include support for LDAP clients/applications. If you plan on developing any applications on this server for use by Notes or Web browser clients, you should choose the Domino Application Server. If you need clustering functionality to provide high availability for your Directory services, you should choose the Domino Enterprise Server. Whichever product you choose, the first step is to install the server software.

---

### 4.1 Installing the Domino server

Consult the documentation that came with Domino, particularly *Setting up A Domino Server*, CT7XKNA, for installation instructions for the Domino product. Once the server software has been installed on the machine, and you have launched the setup process, there are a few items to keep in mind. If you are planning on using this Domino server as a directory server for LDAP-based clients and/or applications, choose **Advanced Setup** in step 2, and make sure that in step 3 of server setup, you check **LDAP** under Internet Directory Services. This will set the LDAP server to automatically run on server startup (see Figure 4 on page 48). Additionally, if you are going to use this server only as a directory server, you can disable the Calendar Connector and Schedule Manager tasks. Finally, if you will be gathering data from other directories that use RDBMS or ODBC-compliant back ends, you might want to enable the DECS service to facilitate that access.

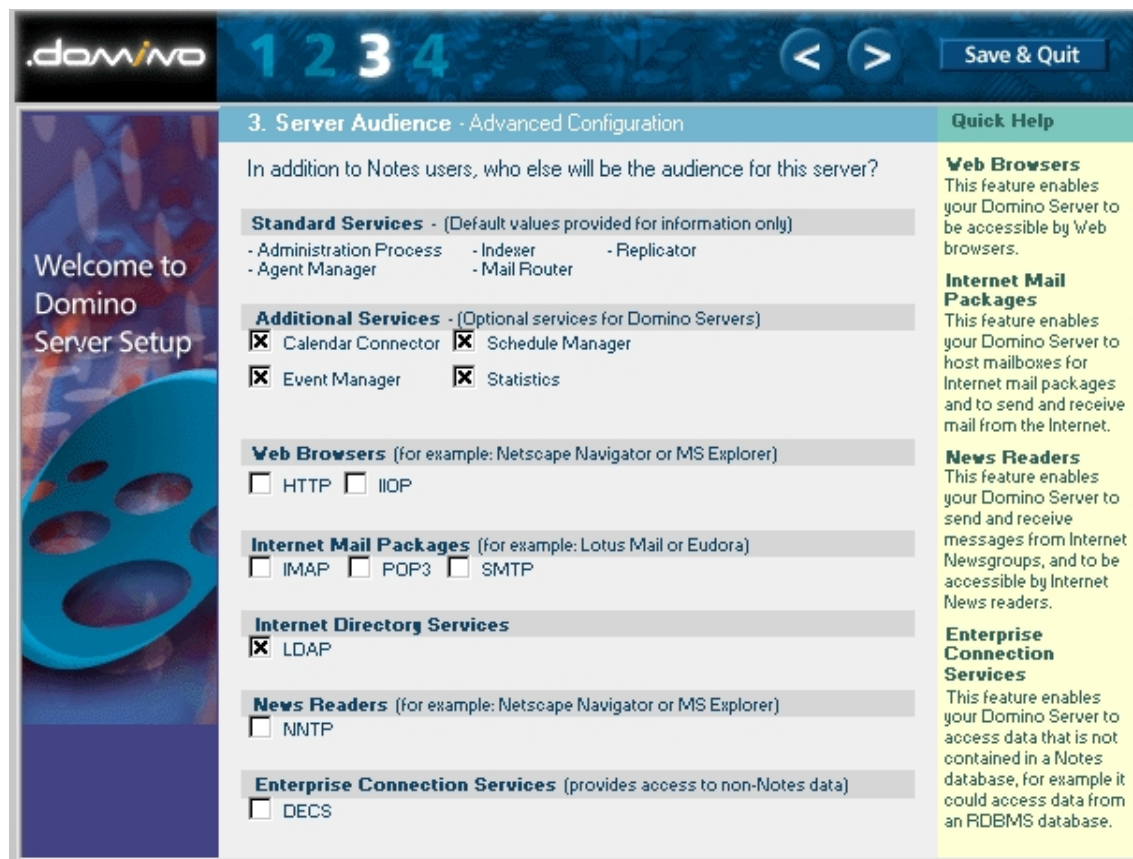


Figure 4. Domino server setup with LDAP selected

## 4.2 Setting up and configuring LDAP services on Domino

Setting up the LDAP service to run on Domino only takes a few minutes. This section describes the steps.

### 4.2.1 Setting up the LDAP service on an existing Domino server

If you are adding LDAP services to an existing Domino server, all you need to do to get the service up and running is type `load LDAP` from a server command prompt. In addition, you will want to modify the "ServerTasks=" line in `notes.ini` by adding "LDAP" to the list of tasks. This will ensure that the LDAP service is automatically loaded at server startup.

**Note**

If you are adding the LDAP service to an existing partitioned server, you will need to add the variable LDAPAddress to your notes.ini file, with the IP address for the partition that is running the LDAP server.

#### 4.2.2 Configuring the LDAP service

To make configuration changes to your LDAP server, you will need to create a server configuration document. From the Administrator client, choose the **Configuration** tab, expand the Server section, and click the **Configurations** icon. Now click the **Add Configuration** action button to create a new configuration document.

Configuration documents that are used by the LDAP server must have the “Use these settings as the default settings for all servers” checkbox selected. Note that there can be only one configuration document that is thus designated. Once you have selected this option, the LDAP tab appears. Click on it to move to that section, and you should see a screen like Figure 5 on page 50.

**Note:** By using the LDAP settings on the default domain configuration settings document, administrators, designers and administrators have a significantly greater assurance that LDAP-based actions will provide consistent results throughout the domain.

Save and Close

## CONFIGURATION SETTINGS

Basics | **LDAP** | Router/SMTP | MIME | NOTES.INI Settings | Administration

### LDAP Configuration

Choose fields that anonymous users can query via LDAP:

Anonymous users can query:

Allow LDAP users write access:	<input type="checkbox"/> No
Timeout:	<input type="checkbox"/> 0 seconds
Maximum number of entries returned:	<input type="checkbox"/> 0
Minimum characters for wildcard search:	<input type="checkbox"/> 1
Allow Alternate Language Information processing:	<input type="checkbox"/> No
Rules to follow when this directory is the primary directory, and there are multiple matches on the distinguished name being compared/modified:	<input checked="" type="radio"/> Don't modify any <input type="radio"/> Modify first match <input type="radio"/> Modify all matches

Figure 5. LDAP configuration settings

Clicking the button will bring up a dialog in which you can add fields to the default fields that anonymous users can query. You can choose from the available forms in the Domino Directory and add fields to those exposed to anonymous users by clicking **Show Fields** and selecting fields to add. You can also remove fields from the list of those available to anonymous users through this dialog. See Figure 6 on page 51.

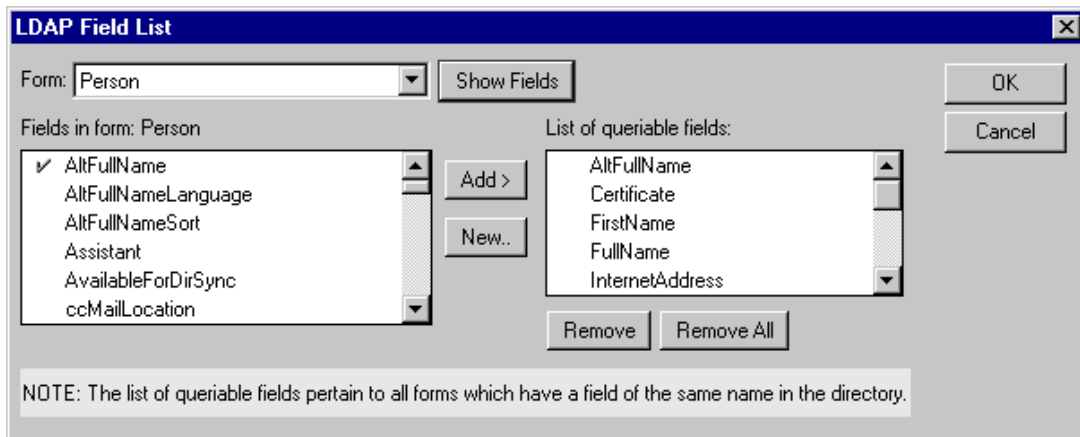


Figure 6. Modifying fields available to anonymous users

“Allow LDAP users write access” is a global setting that enables any LDAP Modify, Add, Delete, and Modify RDN operations to change the Domino Directory. Note that the LDAP service takes full advantage of Domino’s security model. Adding new directory entries requires Editor access in the ACL; modifying existing entries requires Editor or Author (with appropriate modification rights); deleting existing entries requires the “Delete Documents” ability in the ACL. Of course, this setting does not affect anyone’s ability to search the LDAP directory.

The next three settings are for performance tuning the LDAP service.

- Timeout -- specifies the number of seconds to wait before automatically terminating a search request. The default “0” allows for an unlimited time.
- Maximum number of entries returned -- specifies how many results can be returned by a Search operation. Again, “0”, the default, specifies no limit.
- Minimum characters for wildcard search -- allows the administrator to require the user to specify more specifics, thus reducing the result set from a Search operation. The default is “1”.

The next setting determines whether the LDAP service will serve up alternate language information. This information will be UTF8-encoded Unicode. Note that this only works if you have defined alternate language information in the directory. See *Administering a Domino Server*, CT7VHNA and CT7VINA, on how to set up alternate language information for end users.

The last setting governs how the LDAP service handles requests when the DN operated upon is not unique. This applies to modify, delete, and compare operations, or when an add operation is attempted when there is already an

entry in the directory. “Don’t modify any” will prevent the operation from concluding. “Modify first match” will modify, delete, or compare with the first match found. “Modify all matches” will perform the operation on all matches found.

### 4.2.3 Exporting the LDAP schema

To export the LDAP schema to a Notes database, type `tell LDAP exportschema` from the Domino server console. This will create a database, `Schema50.nsf` in the root of the data directory. This database has an entry for every item in the Domino schema and how it maps to the LDAP standard. See Figure 7 for an example of an entry in the schema database.

**Note:** For a discussion of *schema*, see Appendix H, “Directory Schema” on page 253.

Basic		Comments		Advanced	
<b>Names</b>			<b>Syntax</b>		
LDAP name:	cn		Syntax type:	1.3.6.1.4.1.1466.115.121.1.15	
Alternative names:	commonName				
OID:	2.5.4.3				
Notes mapping:	cn				
Schema:	RFC2256				
<b>Matching Rules</b>			<b>Values</b>		
Equality match:			Single valued:	<input type="radio"/> No <input type="radio"/> Yes	
Ordering match:			Collective:	<input type="radio"/> No <input type="radio"/> Yes	
Substrings match:			No user modification:	<input type="radio"/> No <input type="radio"/> Yes	

Figure 7. Document in `Schema50.nsf`

Here you can see the name of the attribute (cn in this case), any alternative names defined in the schema, the object’s OID, the Notes fieldname that is mapped to that attribute, and the Schema from which the attribute is derived.

Note that this database contains Notes fieldnames for attributes that aren’t included on any form, such as `carLicense`. You should check this database before extending the schema, and use the Notes mappings whenever possible before creating your own attributes.



## 4.2.4 Modifying the LDAP Schema

Since the LDAP Schema for Domino Directory services is simply an extension of the inherent design of the Domino Directory, to modify the schema you simply modify that design. All of the design changes are made using the Domino designer client, and are thus made available to both Notes and LDAP clients. The procedure for extending the schema varies slightly depending upon whether you are adding new attributes to the existing schema (person, group, etc.) or adding totally new object classes to the schema.

### 4.2.4.1 Adding new attributes to the existing schema

To ease the addition of new design elements to the existing schema for People, Groups, etc., Lotus has added to the Domino Directory a set of subforms that may be modified to include user-defined fields. These subforms are blank placeholders that can be used to add attributes to the relevant object classes without the risk of losing those changes when the design of the directory is refreshed. The subforms included in the Domino Directory template are:

- \$CertifierExtensibleSchema
- \$DomainExtensibleSchema
- \$GroupExtensibleSchema
- \$MailInDatabaseExtensibleSchema
- \$PersonExtensibleSchema
- \$ResourceExtensibleSchema

You will note that there is not a \$ExtensibleSchema subform for servers. If you find it necessary to extend the server schema, you should add your own subform to the server form. Make sure that the server form is set to “Disallow design refresh/replace to modify” so that your changes won’t be lost. Then add your fields onto the subform. When you upgrade to a newer release and have to update the design of the server form, you should be able to easily reapply your changes in the same way.

We recommend that you create a new subform to store your own extensions, and then insert that subform into the relevant \$extensibleschema subform. You can then make all of your changes in one place. As an example, let’s investigate how you would add a new attribute for eye color to the person object class.

You can make your design changes to either the Domino Directory template (pubnames.ntf) or to the Domino Directory itself (usually names.nsf). In either case, your first step should be to back up the existing copy, in case anything

goes wrong. Next, open Domino Designer and click **Open an existing database...** From the dialog that results, choose the server and the Domino Directory database or template. If you have a multiple-server environment, make sure that you pick a Domino server that can push your design changes to other servers. Your administrative hub server is probably the most logical choice.

Now, create a new subform. From the Design pane, expand the Resources section. Click **Subforms**. Now click **New Subform**. You should be presented with a blank subform. Your first step is to define this subform. Since it is a custom extension of the Person object, we will call our subform “Acme Person.” Choose **Create, Field**. A new field, temporarily named “Untitled” should be created for you on the form, and the Field properties box should appear. Change the field name to AcmePerson. Now set the field type to “Text, Computed when composed.” Close the dialog and move to the programmer’s pane. For the field’s value, enter this formula: “FIELD \$objectclass:=\$objectclass:”AcmePerson”;1” This sets up AcmePerson as an auxiliary object class to dominoPerson.

Now you can add any custom fields that you want. In our example, we are extending the schema to include an attribute for eye color, so position the cursor wherever you want the new field to appear in Notes, and choose **Create, Field**. For our example, we’ll name the field “EyeColor”. This field will be a simple text field. The type of field you create will translate to the type of attribute in the schema displayed in Table 4.

Table 4. Field to schema mapping

Notes Field Type	Schema Attribute Syntax
Text	Directory string
Date/Time	Generalized time
Number	Integer
Names	Distinguished name

This field is a normal Notes field, so you can apply any input validation/translation formulas to it that you wish. Once you’ve added the field, close and save the subform. If prompted, give it the name “Acme Person”.

The final step in adding this element to the person object class is to embed the custom subform that we’ve created (AcmePerson) into the \$PersonExtensibleSchema subform that Lotus has set up for exactly this purpose. To do this, double-click the **\$PersonExtensibleSchema** subform

and choose **Create, Insert Subform**. Now choose the newly-created subform from the dialog that pops up. This will insert it into the \$PersonExtensibleSchema subform. Now close and save the \$PersonExtensibleSchema subform.

If you elected to make your changes to the Directory template database (pubnames.ntf), you will need to refresh the design of that directory from that template in order for your changes to take effect in the Domino Directory. Open the Domino Directory database, then from the File menu do **Database -> Replace Design**, using the template (.ntf) file that has been updated. From Notes, the extra field that you've added will appear in the Person document under the Other tab. If you have a lot of custom data to add to the person form, you could embed a tabbed table containing all of the different fields. You can also add any embellishments to the Notes GUI that you wish to make the appearance better. LDAP clients, of course, will not see the appearance, only the underlying structure.

If you want to see an example of how to write a C program that will accomplish this same task, see 6.3.1, "Using the Notes API" on page 105.

Finally, to recognize the new changes that you've made to the Person object class, you need to reload the schema into the LDAP server. To do this, simply type `tell ldap reloadschema`. This will update the schema with the changes you have made. If you want to verify that your changes have made it into the schema, you can type `tell ldap exportschema` and examine the resulting schema50.nsf database that is created on the directory server.

You can follow those same instructions to extend the other portions of the Domino LDAP schema.

#### 4.2.4.2 Adding new object classes to the schema

To add an entirely new object class to the schema, you need to create a new form in the Domino Directory. Designing forms is basically the same as designing subforms, but they are standalone entities. You won't have to insert them into another design element to make them work.

You will, however, have to add two special fields to the form that you are creating. These will tell Domino where to add the object class into the schema. Once you've created a new, blank form, add a field named Fullname, and set its type to Names. This field will store the Distinguished name of the entry. The second field to add should be called "type", and it should be text, computed when composed. For its formula, use the name of the form. That is, if you are adding a new object class to define printers, for example, name the form "Printer" and give the field "Type" the formula "Printer". The new object

class that you define will be given the value of the field “Type” and will exist in your attribute tree under “Top”.

Now you can add any additional fields to the form. Whenever possible, use the standard LDAP field names from the schema50.nsf database. Use the data type mappings from Table 4 on page 54 to help translate Notes fields to LDAP attributes.

Once you’ve added all of your fields, save the form. You will probably want to create a view in the Domino Directory to display these documents.

---

### 4.3 Setting up and configuring Directory Assistance

The first step in setting up Directory Assistance is to create the Directory Assistance database. From a Notes client, choose **File -> Database -> New...** Click **Template Server** and choose an appropriate server that has the Directory Assistance database template on it. Type an appropriate title for the database, such as Acme Corp.Directory Assistance and a simple filename, such as AcmeDA.nsf. From the list of templates choose **Directory Assistance** and click **OK**. Domino will create the database with your name in the ACL as Manager.

The next step is to create rules in the Directory Assistance database to help servers search the proper directories to find entries. The newly-created Directory Assistance database should be open in Notes. From the Directory Assistance view, click **Add Directory Assistance**. This will bring up a new Directory Assistance document. If you are adding a rule for a secondary Domino Directory, see the next section; if setting up a rule for an LDAP directory, see 4.3.2, “Setting up rules for an LDAP directory” on page 57.

#### 4.3.1 Setting up rules for a Domino Directory

If you are using the Directory Assistance database to refer Notes or LDAP users to a secondary Domino Directory (another company’s Domino Directory, for example), it is a straightforward procedure to tell Domino where to find the directory.

The Basics tab of the Directory Assistance document has information to help administrators organize the rules in the Directory Assistance database. For a Notes rule, the “Domain type” will be “Notes”. “Domain Name” is an identifier that must be assigned to each Directory Assistance rule set. It *must* be unique. It doesn’t have to be the same name as the actual Notes domain described by the directory, but you might want to use that as a guide. It cannot be the same as the primary domain name, unless you are making an

entry for the primary domain Directory, in which case it must be the same. If you enter a domain name that is not unique in the Directory Assistance database, you may get the error “User not found in any Name and Address Book” when trying to search the secondary directory. “Company” is a field that can be used to keep track of domains whose names don’t obviously point to one company. “Search Order” will determine the order in which each directory is searched for matches. “Enabled” provides a mechanism for quickly enabling and disabling lookups to foreign directories.

The rules that define when Directory Assistance should consult another directory are specified on the second tab (Rules). Once you’ve finished filling out the Basics tab, click on Rules and describe what scenario applies to this directory. If you want this directory to be searched regardless of the name given for a match, you can put in a rule with all asterisks. More likely, a given directory will hold information for people in given organizations, so you can fill in the organization field. If you want to include a specific OU, you can list it in the relevant OU field(s), you can use a wildcard (\*) character to include all OUs, or you can leave the field blank to prevent matching entries with an OU in that position.

After you define your matching rules, you can specify whether that rule is enabled (again, to provide an easy mechanism to quickly enable/disable matching), and whether or not it can be used for authentication to the server (“Trusted for credentials”). A rule that is not trusted can still be used for mail addressing from a Notes client, and so forth, but will not be used to verify credentials for authentication.

The final tab of the Directory Assistance document, Replicas, tells the Domino server where to look for the secondary directories. You can either use database links or server and pathnames. If you have a replica of the database on all servers that have Directory Assistance, you can use an asterisk (\*) in the server field to tell the Domino server to look locally for the directory.

#### **4.3.2 Setting up rules for an LDAP directory**

The other option when creating a Directory Assistance document is to create one for an external LDAP server. This can be used by Notes clients to look up mail addresses, by Web browsers to authenticate to a Domino server using an external directory, or by LDAP clients and/or applications looking for LDAP information, which will receive a referral to the new directory.

Most of the Basics tab remains the same when you choose LDAP for the Domain Type. Domain name, company name, search order, and enabled all perform the same function for an LDAP directory as a Domino Directory. The

additional field, “Group Expansion”, allows you to specify one LDAP directory that Domino will use for groups. This setting can only be used for one LDAP directory.

The Rules tab operates in the same manner as for an external Domino Directory. The LDAP attributes ou, o, and c map to the Domino attributes OU, O, and C, respectively, so it’s easy to set up the rule. Note that LDAP rules can only be used to authenticate Web users to a Domino Web server, not Notes users.

The LDAP tab of the Directory Assistance document is where you can specify the LDAP-specific parameters for the rules defined in this Directory Assistance document. You first need to specify the hostname of the server that is running the LDAP server. In addition, if the external server requires a username and password to bind to it, enter those in the next two fields. We recommend that you encrypt the fields containing the username/password combination so that they won’t be stored in the clear. See the following note for more information on how to do this. Many LDAP servers require you to provide a base DN (such as “o=Lotus,c=US”) to search upon. If your LDAP server requires this, enter the base DN in the field provided.

#### Encrypting fields

To encrypt the Directory Assistance document and thereby restrict access to it, choose **File ->Document Properties**. Click the **Security** tab (the one with the little key icon) and choose the people who should have access to the document by clicking on the person icon next to Public Encryption keys.

The next two check boxes describe what clients you want to be allowed to use this directory. Notes clients (for mail lookup) and Web clients (for authentication to Domino servers) are selected by default. If you want to allow LDAP clients to use the external directory (by way of a referral) you should check that box as well.

In order to secure the channel between the Domino server and the external LDAP server, it is recommended that you enable SSL in the next field. This will encrypt the packets sent across the wire between the two servers and prevent anyone from capturing directory data. If you want your Domino server to not connect when the server on the other end has an expired certificate, select that option. Unless you have unusual difficulty getting the Domino server to negotiate the SSL protocol with the external LDAP server, you should probably choose the Negotiated setting for the SSL protocol version.

This will allow the two servers to freely negotiate what SSL level to use to secure the channel. Finally, you can add one more layer of security by requiring that the SSL certificate presented by the server match its domain name by selecting Enabled in the “Verify server’s name with remote server certificate” field.

The last two fields on this form are for performance tuning purposes. Here you can set the timeout for queries to the external directory (the default is 60 seconds). You can also set a maximum number of entries returned (default is 100). If you are having trouble getting timely response from the external server, you may want to adjust these settings.

### 4.3.3 Deploying Directory Assistance

Once you have set up your matching rules, you will need to make the Domino server aware of the Directory Assistance database. To set that up, you will need to add its filename to the server’s Server Document in the Domino Directory. To do this, from the Administrator client, choose the **Configuration** tab, then expand the Server section in the left-hand pane and click the **Current Server Document** icon. On the Basics tab, you will see a place to enter the filename of the Directory Assistance database. Enter the name, save the document, and then reboot the server.

If you have additional servers upon which you want to place the Directory Assistance database, simply make a new replica of the database on each server and follow the same procedure. Remember to pay attention to network issues, especially if you have slow links between the servers using Directory Assistance and the servers holding replicas of the secondary directories. Also consider that different servers can use different Directory Assistance databases, if you want a different set of directories for a Web server, for instance.

---

## 4.4 Setting up and configuring Directory Catalogs

When creating your Directory Catalog, the first choice you will need to make is how you want the catalog to be deployed. A Directory Catalog can be placed either on the server, so that all users can use it for type-ahead and address resolution, and so that the server can use it to speed the process of authentication for Web users; or it can be pushed down to the clients (a Mobile Directory Catalog) so that each client can have an entire enterprise directory, even when disconnected from the network. The two are not mutually exclusive; you can use a server Directory Catalog to improve server mail routing performance while deploying a mobile Directory Catalog to end

users, particularly those who will be operating in a disconnected mode at times. In this section, we'll investigate how to set up both possibilities.

#### 4.4.1 Setting up a server Directory Catalog

Before you set up a Directory Catalog, make sure that your directory profile is set up correctly. To do this, open the directory and choose **Actions -> Edit Directory Profile**. Make sure that there is an entry in the "Domain defined by this public directory" field, and that it is unique among directories that will be aggregated. If this directory is used for Notes mail users, it should be the same as the mail domain for those users. You will need to take this step for every directory that is going to be aggregated. If you have directories that are using an older version of Domino, you should upgrade the design of those databases to the R5 Domino Directory template. If you do not have a directory profile, you will encounter difficulties if groups in more than one directory have the same name. Domino will not be able to distinguish between the two and your mail might get routed to the wrong set of people.

Additionally, you will need to make sure that you have replicas of every directory to be aggregated on the server. While it is possible to have the Directory Catalog gather entries from remote servers, it is not recommended. Creating the catalog will take significantly longer. Remember to create connection documents back to the servers you replicate the other directories from, to ensure that the data is kept up-to-date.

Now you need to create the database that will hold the aggregated directories. From a Notes client, choose **File -> Database -> New...** Choose the server you will be using to aggregate the directories. Type in a descriptive name for the Directory Catalog, such as Acme Directory Catalog (for a server-based Directory Catalog) or Acme Mobile Directory Catalog (for a mobile Directory Catalog). Type in a concise filename, such as acmedircat.nsf.

Click **Template Server...** and choose a server with the **DirCat** template, select the **Directory Catalog** template (being careful not to select Directory Assistance or Catalog (5.0) instead) and click **OK**. Select **Yes** when prompted to create a full text index for the Directory Catalog. A full text index is especially needed if this server will be handling LDAP queries.

#### 4.4.2 Configuring the Directory Catalog

Now that you've created the database, open it and choose **Create -> Configuration**. In this form, specify what directories you want to aggregate



into this Directory Catalog, and some additional configuration parameters. The form you see should look like Figure 8.

**DIRECTORY CATALOG CONFIGURATION**

Basics | Advanced

**Basics**

Directories to include:

Additional fields to include:  
(Fullname and ListName included by default)

☐ FirstName  
☐ MiddleInitial  
☐ LastName  
☐ Location  
☐ MailAddress  
☐ Shortname  
☐ MailDomain  
☐ InternetAddress  
☐ MessageStorage

Sort by:

☒ Distinguished Name  
☐ Last Name  
☐ Alternate Fullname

Use Soundex:

Remove duplicate users:

Group types:

Include Mail-in Databases

Restrict aggregation to this server:

Send Directory Catalog reports to:

Comments:

Figure 8. Directory Catalog configuration: basics

In the first field, “Directories to include”, list the filenames of the directories that you want to aggregate into this Directory Catalog. It is strongly recommended that you have replicas of each directory on this server. If not, you can use the syntax `port!!!server!!filename` to include directories on foreign servers, for example `tcpip!!!hubserver1/acme!!acmedir.nsf`.

The next field controls what fields from the Domino directories are added to the Directory Catalog. The Directory Catalog includes FullName (for people) and ListName (for Groups) by default. If you want to include any other information, make sure that the field containing that information is included in the configuration. Note that one benefit of the Directory Catalog is its small size, so you probably wouldn’t want to include any of the certificate or public

key fields from the Domino Directory, since they would take up so much space. Remember that users will still be able to send encrypted mail from their workstations using just-in-time encryption. One field you probably do want to include, at least for server-based Directory Catalogs, is the Members field from group documents. This will allow the server to retrieve the group's member list from the Directory Catalog rather than the source directory. This is not an issue for mobile Directory Catalogs, since Group expansion will be handled by the server anyway.

Another method the Directory Catalog uses to reduce size is to only have one indexed view. View indices in Domino make for quick, efficient searches, but they can consume a lot of disk space. Therefore, the next field on the configuration document lets you pick what sort of index you want to create. This is particularly important for type-ahead. Distinguished Name is the default; keep this selection if your end users prefer typing first names to get resolution. If your end users prefer to type last name first to get resolution, or want to use alternate language names, you will need to choose that option here. You will also need to make sure that you haven't removed those fields from the "Additional fields to include" setting.

If you want the Directory Catalog to include Soundex information, so that users can slightly mis-type the name and still get suggested matches, enable this feature in the next field. Again, this will slightly increase the size of your Directory Catalog, so if space is your primary concern, you might want to disable it.

"Remove duplicate users" will consider the first record for a given person to be the authoritative one, and won't include any entries encountered later. This is a good option to use if your secondary directories contain duplicate entries that aren't used for mail routing or authentication. Note that the order of filenames in the "Directories to include" field is the order in which they will be added to the Directory Catalog, so make sure your most authoritative directories are listed first. Also be aware that this setting can cause headaches if person documents are removed from the primary directory, and remain in secondary directories. Since the entries in the secondary directories haven't changed, they won't automatically be included when the Directory Catalog is next aggregated. You'll have to modify the secondary entries to get them included the next time the aggregator runs.

The "Group Types" setting allows you to select which types of groups you wish to include in the Directory Catalog. The default is "Mail and Multi-Purpose". If you need to include other types of groups (if you're authenticating Web users by group membership and you want to speed up the process of getting the Members list, for example), you can choose other

options here. In most cases, the default setting will eliminate groups not needed, thereby trimming the size of the Directory Catalog. Similarly, the “Include Mail-In Databases” setting will allow you to remove mail-in databases if you do not wish to include them in the Directory Catalog.

The “Restrict aggregation to this server” setting allows you to specify the single server that will aggregate directories into this Directory Catalog. If you put a server name into this field, attempting to aggregate this Directory Catalog on another server will almost definitely generate replication conflicts. If there is an entry in this field, attempting to run the aggregator task on another server against this Directory Catalog will generate the error “Aggregation of this catalog can only be done by <servername>” and the aggregation task will fail. This setting provides an extra level of security against database corruption or conflicts arising from multiple servers modifying the same Directory Catalog, then replicating with each other.

“Send Directory Catalog reports to” should be a list of names to receive weekly reports from the Directory Catalog Status Report agent. See 5.1.1, “Monitoring the Directory Catalog” on page 69 for more information.

Switching to the Advanced tab, shown in Figure 9, you will see parameters to improve performance in the Directory Catalog.

Figure 9. Directory Catalog configuration: Advanced

If you want to use a secondary Domino Directory only for certain people or groups, you can apply a selection formula that the Directory Catalog will use when determining what entries to aggregate. For example, if you only wanted

to include people who have Notes Mail on a given server, you could use a formula such as:  
`SELECT @lowercase(MailServer)="cn=mailhub1/ou=servers/o=acme"`

Similarly, you could use any other field in the Person or Group document to use as the basis for your selection formula. Remember that the selection formula is applied to all entries in all directories, so make sure that the formula applies equally to all different directories aggregated in this Directory Catalog.

In order to conserve space, the Directory Catalog creates aggregate documents from multiple individual documents in aggregated directories. The "Packing density," "Incremental fields," and "Merge factor" settings govern how the Directory Catalog creates these documents.

"Packing density" determines how many directory documents are aggregated into a single document in the Directory Catalog. The default (and the maximum) is 255. You would only want to lower this number if there are a lot of searches using the full text index to resolve names, rather than by way of the indexed view. LDAP searches always use the full text index, so you might want to decrease this setting in that scenario. Lowering this setting will increase the size of the Directory Catalog.

"Incremental fields" and "Merge factor" govern how the Directory Catalog minimizes replication. If you have 255 documents in a source directory that are aggregated into one document in the Directory Catalog, you are 255 times more likely to have to replicate that single document than in a traditional directory. The Directory Catalog minimizes this problem by consolidating all changes into "Incremental fields" on those documents, then changes the "permanent fields" when a threshold of changes is met, determined by the "Merge factor" setting. This significantly reduces the amount of replication traffic generated, but does slightly impact search performance. Be sure to consider network links between replicas of the Directory Catalog if you plan on changing these settings.

Once you have made all of your configuration settings, save and close the configuration document. You are now ready to build the Directory Catalog.

#### **4.4.3 Running the Directory Catalog aggregator**

On the server with the Directory Catalog database, type `load dircat <filename>` on the server console. This will begin the process of building the Directory Catalog. It will take about 1 hour for every 75,000 entries to be aggregated on a Pentium 200-class server. On our test server, a 350 MHz

Pentium II with 192 MB RAM, we were able to aggregate 3 directories, with a total of approximately 11,000 entries, in 3.5 minutes. Keep in mind that changing the fields included or the number of directory documents aggregated into single documents will affect the amount of time it takes to build the Directory Catalog. Once the Directory Catalog is built, you should update the full text index by typing `load updall <filename>` from the server console. The Directory Catalog makes extensive use of the full text index, so don't forget this step.

#### 4.4.4 Scheduling the Directory Catalog aggregator

Once your Directory Catalog has been created, you should schedule updates to keep the information current. This is set up in the server document for the directory server. To change these settings, from the Administrator client go to the Configuration tab, expand the Server section, and click on "Current Server Document". When the server document is open, Click the "Edit Server" action button. Now switch to the Server Tasks tab and then click the Directory Cataloger tab. You should see a screen similar to the one shown in Figure 10.

Figure 10. Directory Catalog configuration in the server document

In the Basics section, list the filenames of the different Directory Catalogs aggregated on this server. If you have more than one, separate them with commas. In the schedule section you can set up a schedule for when and how often the directory aggregator will run. If this server is primarily used for directory services, you can stick with the default schedule, or even increase the frequency. If this server performs other operations, you may want to

schedule directory aggregation for after hours or less frequently during the day. Remember that you do not want to run the aggregator task against a Directory Catalog on more than one server.

#### 4.4.5 Making the Directory Catalog available to the server

The final step in setting up a server-based Directory Catalog is to make it available to the server. If you are deploying this Directory Catalog onto every server in your Domino environment, you should edit the Domino Directory Profile. To do this, open the Domino Directory and choose **Actions -> Edit Directory Profile**. Move to the Directory Catalog database name for domain field and type the name of the Directory Catalog.

If you want to set up the Directory Catalog only on certain servers, or if you want to use different Directory Catalogs on different servers, you should edit the individual server documents. To do this, use the Domino Administrator client. First, make sure that the server you want to use the Directory Catalog is chosen, then choose the **Configuration** tab, expand the **Server** section, then click **Current Server Document**. Click the **Edit Server** action button to make your changes. Now move the cursor down to the Directory Catalog database name on this server field and type in the filename of the Directory Catalog. Click **Save** and **Close**, then reboot the server to have this change take effect.

#### 4.4.6 Setting up and configuring a mobile Directory Catalog

The process for creating a mobile Directory Catalog is basically the same as for the server-based Directory Catalog. However, for a mobile Directory Catalog, it is not necessary to add its filename to the server document or the domain directory profile. Otherwise, the steps are the same. Even though the Directory Catalog is not being used by the server itself, remember to schedule it for aggregation along with server-based Directory Catalogs. This will ensure that the directory information is kept up-to-date.

#### 4.4.7 Deploying a mobile Directory Catalog to end users

To get a mobile Directory Catalog deployed to a workstation, you have three choices:

- Send the user a detailed note explaining how to create a local replica of the Directory Catalog and how to update their mail preferences under the user preferences settings. If you want to deploy it to every user in the organization, and you do not have a sophisticated user community, this is not a viable option.

- The same process can be automated by sending the user an e-mail where the form is designed to create the replica and change the names statement in the notes.ini file. In a user community where some of the workstations might not have enough space to receive and maintain a mobile or user Directory Catalog, this might be the preferred option.
- The easiest option to take is to make use of user profile documents. You will need to create a user setup profile with a doclink of the mobile Directory Catalog in the document. This Directory Catalog will then be pushed to the workstation at the next server connection, but only if the workstation is an R5 client.

To create a Setup Profile document, launch the Domino Administrator and click the **People & Groups** tab. Now click the **Setup Profiles** icon. To create a new setup profile, click **Add Setup Profile**. You should see a form like the one shown in Figure 11.

The screenshot shows the 'USER SETUP PROFILE' form. At the top, there is a title bar 'USER SETUP PROFILE' and a series of tabs: Basics, Databases, Dial-up Connections, Accounts, Name Servers, Applet Security, Proxies, MIME, and Administration. The 'Basics' tab is currently selected. Below the tabs, the 'Basics' section contains several fields: 'Profile name:' with a text input field; 'Internet browser:' with a dropdown menu showing 'Notes'; 'Directory server:' with a text input field; 'Catalog/Domain Search server:' with a text input field; 'Retrieve/open pages:' with a dropdown menu showing 'from Notes workstation'; and 'Sametime server:' with a text input field.

Figure 11. Setup profile form

Enter a descriptive name in the Profile name field. Click the **Databases** tab and move down to the Mobile Directory Catalogs field. In this field, you can paste a database link to the mobile Directory Catalog that you have created on the server. Click **Save & Close** to save this profile.

To associate a Setup profile with a user, you need to modify his person document in the Domino Directory to point to the Setup Profile that you've created. To do this, switch to the **People** view, select the person you want to associate with the profile, and click **Edit Person**. On the Administration tab move to the Setup profile(s) field. Here you can type the name of the profile to be used. Once you've added the profile name, Save and Close the person document.

The next time the client connects to the server, it will make a local replica of the Directory Catalog in its local data directory, create its full-text index, and modify its preferences to include the local replica in the Address Books preference. Note that, unlike a server, a workstation can use multiple Directory Catalogs.



---

## **Chapter 5. Administering Domino Directory services**

This chapter provides advice to administrators responsible for Domino Directory services. We discuss monitoring, performance tuning, and troubleshooting Domino Directory services, and finish up with some best practices for Domino Directory services deployment.

---


### **5.1 Monitoring Domino Directory services**

Keeping informed about the operation of Domino Directory services is critical to the system administrator. In a complex directory environment, there are many different tasks and processes that are working on directories, and a failure in any one of them might not easily be noticed. Fortunately, Lotus provides many tools to help the administrator stay informed about these tasks.

#### **5.1.1 Monitoring the Directory Catalog**


One of the most useful tools that is provided to you for monitoring the Directory Catalog is the status report agent in the Directory Catalog database itself. When you configure the catalog, put a group of people that should be updated on its status into the “Send Directory Catalog reports to:” field. Those people will receive an e-mail update every week with the status of the Directory Catalog. See Figure 12 on page 70 for an example of the memo that is generated.


```


Redbook Directory Catalog >>  <<

-- Properties --
Title           : Redbook Directory Catalog
Server          :
Path            : c:\lotus\domino\data\RedBookDirCat.nsf
Size            : 2,621,440 bytes
Total Entries   : 10,994.00

-- Configuration --
Version         : 3
Sort Key        : Distinguished Name
Use Soundex     : Yes
Remove duplicates : Yes
Group Types     : Mail and Multi-purpose
Packing Density : 255
Selected Fields : FirstName, MiddleInitial, LastName,
Location, MailAddress, Shortname, MailDomain,
InternetAddress, MessageStorage,

-- Directories to Include --
RedBook's Address Book >>  <<
Server          :
Path            : names.nsf
Size            : 4,980,736 bytes
Updated in DirCat : 7/28/00 5:11:34 PM

IBMAT's Address Book >>  <<
Server          :
Path            : AT_names.nsf
Size            : 14,417,920 bytes
Updated in DirCat : 7/28/00 5:11:35 PM

IBMNL's Address Book >>  <<
Server          :
Path            : NL_names.nsf
Size            : 27,000,832 bytes
Updated in DirCat : 7/28/00 5:12:28 PM

-- Report --
Total size of Source Directories is: 46,399,488 bytes
Redbook Directory Catalog is 5.65% of the size of the
directories used to create it.

```

Figure 12. Directory Catalog statistics report

If you want to see more information about the Directory Catalog aggregator task, you can enable detailed logging by putting the variable `log_dircat=1` into your `notes.ini` file. Following is some sample output from a Directory Catalog operation with the `.ini` setting in effect:

```

07/28/2000 05:11:33 PM Directory Catalog Needs rebuilding (Note: replication
to all mobile clients will take longer)
07/28/2000 05:11:34 PM Directory Cataloger processing directory names.nsf in
catalog redbookdircat.nsf
07/28/2000 05:11:34 PM Directory Catalog directory 'names.nsf' uses 'RedBook'
Domain for Groups
07/28/2000 05:11:35 PM Directory Cataloger Finished Processing names.nsf
07/28/2000 05:11:35 PM ...6 Add(s)
07/28/2000 05:11:35 PM Directory Cataloger processing directory AT_names.nsf
in catalog redbookdircat.nsf
07/28/2000 05:11:35 PM Directory Catalog directory 'AT_names.nsf' uses 'IBMAT'
Domain for Groups
07/28/2000 05:12:28 PM Directory Cataloger Finished Processing AT_names.nsf
07/28/2000 05:12:28 PM ...3418 Add(s)
07/28/2000 05:12:28 PM Directory Cataloger processing directory NL_names.nsf
in catalog redbookdircat.nsf
07/28/2000 05:12:28 PM Directory Catalog directory 'NL_names.nsf' uses 'IBMNL'
Domain for Groups
07/28/2000 05:13:36 PM Searching Administration Requests database.
07/28/2000 05:14:50 PM Directory Cataloger Finished Processing NL_names.nsf
07/28/2000 05:14:50 PM ...7570 Add(s)
07/28/2000 05:14:50 PM Directory Cataloger finished processing Directory
Catalog redbookdircat.nsf
07/28/2000 05:14:51 PM Directory Cataloger process shutdown

```

You can also see the time that the Directory Catalog aggregator is next scheduled to run by using the `show schedule` command on the Domino server console. You should see output like this:

```

> sh sch
Directory Cataloger          Directory Cataloger          08/07/2000 02:00:00 PM

```

### 5.1.2 Monitoring the LDAP service

The Domino server is constantly updating usage statistics for the LDAP server. These are under the LDAP heading. Here's a sample of the output from a `show stat LDAP` command:

```

LDAP.Sessions.Accept.Queue = 0
LDAP.Sessions.Active = 1
LDAP.Sessions.Inbound.BytesReceived = 432
LDAP.Sessions.Inbound.BytesSent = 4,051
LDAP.Sessions.Inbound.non-SSL = 2
LDAP.Sessions.Inbound.Total = 2
LDAP.Sessions.Peak = 1
LDAP.Sessions.Threads.Busy = 1
LDAP.Sessions.Threads.Idle = 0
LDAP.Sessions.Total = 2
LDAP.Simple LDAP Connections = 2
LDAP.Total LDAP Connections = 2
LDAP.Total LDAP Search Entries Returned = 11
LDAP.Total LDAP Searches = 6

```

Table 5 provides a brief description of the LDAP statistics.

*Table 5. LDAP statistic definitions*

<b>Statistic</b>	<b>Explanation</b>
LDAP.Anonymous LDAP Connections	The number of anonymous connections to the LDAP server. Chances are that most of your connections will be simple searches from anonymous LDAP users.
LDAP.Sessions.Accept.Queue	The number of sessions that are currently waiting to be accepted. If this statistic seems high, consider adding more system resources, setting up a separate LDAP server to handle the load, or removing other processes from the server.
LDAP.Sessions.Active	The number of currently active LDAP sessions.
LDAP.Sessions.Inbound.BytesReceived	The total number of bytes received from LDAP clients.
LDAP.Sessions.Inbound.BytesSent	The total number of bytes sent to LDAP clients.
LDAP.Sessions.Inbound.non-SSL	The total number of sessions that haven't used SSL.
LDAP.Sessions.Inbound.SSL	The total number of sessions that have used SSL.
LDAP.Sessions.Inbound.Bad_Handshake	The total number of sessions that have been attempted with SSL, but have failed due to handshake problems. This is probably due to non-matching certificates.
LDAP.Sessions.Inbound.Total	The total number of LDAP connection attempts.
LDAP.Sessions.Peak	The peak number of sessions since the LDAP server was last restarted. Compare this number to LDAP.Sessions.Active to see how your current usage compares to historical highs.
LDAP.Sessions.Threads.Busy	How many threads are currently in use by the LDAP server.
LDAP.Sessions.Threads.Idle	How many idle threads are allocated to the LDAP server.

Statistic	Explanation
LDAP.Sessions.Total	The total number of LDAP sessions since the LDAP server was last started.
LDAP.Simple LDAP Connections	The total number of authenticated LDAP connections (non-SSL).
LDAP.Total LDAP Adds	The total number of additions to the Domino Directory made via LDAP.
LDAP.Total LDAP Connections	The total number of connections made to the LDAP server. This number shows the actual connections made to the LDAP server, as opposed to attempted connections, which are reflected in the .sessions statistic.
LDAP.Total LDAP Deletes	The total number of deletions from the directory made via LDAP.
LDAP.Total LDAP Modifies	The total number of modifications to the directory made via LDAP.
LDAP.Total LDAP ModifyDNs	The total number of modifyDN operations made to the directory via LDAP.
LDAP.Total LDAP Referrals Returned	The number of referrals that have been passed to LDAP clients for searches in secondary LDAP directories.
LDAP.Total LDAP Search Entries Returned	The total number of entries returned to LDAP searches.
LDAP.Total LDAP Searches	The total number of LDAP searches conducted against the directory.

It is also possible to set up a probe to make sure that the LDAP server is running and return some information about response time. To do this, from the Administrator client, click the **Configuration** tab, then expand the **Statistics & Events** section and the **Probes** folder. Click **TCP Server** and then click the **New TCP Server Probe** action button. You should see a new form like Figure 13 on page 74.

**TCP Server Probe**  
Event Number: BALR-4MTHTG

Basics | Probe | DNS | FTP | HTTP | IMAP | LDAP | NNTP | POP3 | SMTP | Other

☐ All Domino servers in the domain will probe their own configured ports

**Target Server(s)**

☒ All in the domain  
☐ Only the following:

**Probing servers (source)**

Server(s):  Servers...

Figure 13. TCP server probe form

On the Basics tab, click **Only the following** and choose the directory server you wish to monitor. On the Probe tab, modify the Interval and Timeout settings if necessary, then click **Probe these services** and deselect everything except LDAP. The LDAP tab will show you the statistic name that is going to be generated. Click the **Other** tab and **Generate a new notification profile for this event**. This will start the event notification wizard.

On the first screen that appears, select **A custom monitor or probe event**. Click **Next**. Choose a notification method and a schedule, if you wish, and click **Next**. Depending upon the method you chose, give the wizard some details about who or how you wish to be notified. Click **Next**, then click **Finish** to set up your event notification. The probing server will now send a probe at whatever interval you've specified, and if it fails you will be notified by whatever method you chose.

The value of the statistic that is returned from the probe will be the response time of the LDAP server. This can be used to measure trends and plan for additional resources as the response time slows.

---

## 5.2 Performance tuning Domino Directory services

There are many ways to tune Domino server performance in general. Since the Domino Directory is, at its heart, just another Domino database, any of those performance tuning methods will improve directory performance.

Don't load any server tasks that aren't needed. For a server that is just handling directories, you can probably quit every task except for Replica, Update, AMgr (for DirCat status agents), Event (for monitoring the server), and LDAP. Carefully go through the list of server tasks in your notes.ini and prune out those that you don't need.

Remember that the Domino Directory is a database just like any other Notes database. Any performance enhancements you make to the entire server, such as implementing transaction logging or moving view rebuilds to a separate drive, will increase the performance of the directory.

### 5.2.1 Performance tuning for Notes users

One big way that you can improve client performance is to deploy a mobile Directory Catalog. Even if the client workstations are connected to the network, you can push a copy of the Directory Catalog to the end users' workstations. Clients using this directory will not hit their mail server for type-ahead lookups. Furthermore, you can disable type-ahead completely on the server by adding a notes.ini variable.

You can also increase performance by separating addressing and type-ahead issues from mail server operations. To do this, specify a separate directory server in the client's Location document, on the Servers tab. This will allow name lookups/type-ahead to be done against a centralized server that is specialized for that task, while mail reads/writes, routing, and so forth operate against the user's normal mail server. To ease administration, you can push this setting down to client workstations with a setup profile. See 4.4.7, "Deploying a mobile Directory Catalog to end users" on page 66 for more information about setup profiles.

### 5.2.2 Performance tuning for LDAP users

Most of the performance tuning for the LDAP server is done in the LDAP portion of the Server Configuration document (see Figure 5 on page 50). Specifying a value other than 0 in the Timeout field is an easy way to prevent LDAP searches from consuming too much of the server's resources. Another method is to limit the number of hits returned from an LDAP search, using the "Maximum number of entries returned" setting. Specifying the minimum

number of wildcard characters will also prevent searches from returning too many results.

Remember that LDAP searches of Directory Catalogs rely extensively upon full text indices. Therefore, any performance improvements in this area will be especially useful to LDAP users. If space is not at a premium on the LDAP server, decreasing the packing density of the Directory Catalog can speed up searches. Since the Directory Catalog saves space by both pruning out unnecessary fields and by combining multiple documents from the source directory into one document in the catalog, you can lower the packing density, even as low as 10, and still save a lot of space. Try changing that parameter and see if it helps server performance. An additional benefit of a lower packing density is fewer replication events.

### **5.2.3 Performance tuning for address lookup**

From a mail addressing perspective, you can gain an impressive performance boost simply by separating your directory servers from your mail servers. While it is not at all uncommon to have separate mail and application servers, many organizations are just now starting to realize the benefit of having separate, centralized directory servers. In the Notes R5 client location document, there are separate fields for Mail and Directory server. By moving your name lookup traffic to a separate server, you can improve performance for mail routing and mail database operations, while also providing a centralized repository for all directory data.

The other big impact, for address lookup, is to use a mobile Directory Catalog. By putting the load of lookup and type-ahead onto the client, and by using a Directory Catalog, you can have near-instantaneous type-ahead and address resolution, even in a Directory Catalog that has hundreds of thousands of names in it.

Once you have moved the address lookup off of your mail servers, by using separate directory servers and/or by pushing mobile Directory Catalogs down to end users, you can disable typeahead on the mail server by changing the server configuration document. On the Basics tab of the Server Config document, set Type-ahead to disabled.

### **5.2.4 Performance tuning for authentication**

If your directory is serving primarily as an authentication point for various clients, you can take some concrete steps to improve that performance.

Whenever possible, make sure that secondary directories trusted for authentication are present on the server handling the authentication request.



In fact, consolidating those directories into a Directory Catalog can speed the retrieval of passwords for Web users. Remember to still include rules in Directory Assistance for the secondary directories that are aggregated in the catalog; otherwise, authentication will not work.

---

## 5.3 Troubleshooting Domino Directory services

There are inevitably going to be times when Domino Directory services don't work as expected. Perhaps a name that you know exists in a secondary directory isn't being resolved properly. Perhaps users from a secondary directory aren't being authenticated for access to your Web site. Whatever the symptom, here are some troubleshooting tips to help get to the bottom of the problem.

### 5.3.1 Troubleshooting the Directory Catalog

If names are missing from the Directory Catalog, check the following possibilities:

- The server doesn't have access to all of the directories listed.

This is usually not a problem when the server has local replicas of all of the databases that are listed.

- The DirCat task did not complete its last aggregation.

The first place to check should be the Directory Catalog configuration document. In the field labeled Replication History, check the number and times of date entries. There should be one entry for each directory that is aggregated. If you are unsure, set the notes.ini variable log\_dircat=1 and load the dircat task manually to see if there are any problems.

- The user has been deleted from the primary directory.

If you've enabled the option "remove duplicate users" and a user has been removed from the primary directory but still remains in a secondary directory, you will need to modify the entry in the secondary directory to get the Directory Catalog to include that entry.

- The user doesn't have an entry in the FullName field.

The Directory Catalog will only aggregate people that have entries in the FullName field. If you've manually created people without one, you'll have to add something in that field to ensure they are aggregated.

If type-ahead addressing isn't working as expected, make sure that the "sort by" format that is configured for the Directory Catalog is what is being used to enter the address. If the setting is "Distinguished name," the lookup will look

for a first name to match what's been entered. When it's set to "Last name", it will look for a last name.

If Domino is not finding any entries in the Directory Catalog, make sure that the server has a replica of the Directory Catalog, and that either the Public Directory Profile or the server's individual server document specifies the correct filename of the Directory Catalog.

If LDAP queries against a Directory Catalog aren't working, make sure that the Directory Catalog has a full text index. This is required for LDAP searches.

If the directory server is returning multiple matches for names, even though you've disabled the "exhaustive lookup" option, be aware that the Directory Catalog is checked for addresses even if "exhaustive lookup" is not selected. Make sure that all entries for the same person are the same in both the primary directory and the Directory Catalog.

If you are using a Directory Catalog to look up passwords for Web users, and authentication doesn't seem to be working properly, check the following items:

- Make sure authentication works without the Directory Catalog.  
Remove the Directory Catalog from the server's server document or Directory Profile. Make sure that there are authentication rules in place in the Directory Assistance database. Restart HTTP and check authentication against the secondary directories. If this fails, there's probably an issue with the Directory Assistance configuration.
- Make sure you've included the HTTP Password field in the Directory Catalog.  
There's little point in using the Directory Catalog for authentication if Domino has to go to the source directory to find the password.
- Make sure users are using the correct name to log in.  
If you've enabled the "Fewer name variations with higher security" option in your Web server configuration, make sure that users are logging in with their common names or hierarchical names, rather than short names, first names, or last names. Also make sure that the secondary Domino directories are using the R5 design.

### 5.3.2 Troubleshooting Directory Assistance

Be aware that external LDAP directories don't show up in the Mail Address dialog, nor do they work for type-ahead. Use the F9 key to test resolution of addresses against an external LDAP directory before sending a message.

Make sure you are not listing a Directory Catalog in the Directory Assistance database as a secondary directory. The Directory Catalog is not designed to be used as a secondary directory in this manner.

Make sure the domain names used in Directory Assistance are unique. If there is a conflict, you will receive the "User not found in any Name and Address Book" error.

### 5.3.3 Troubleshooting LDAP lookups against external directories

If you are having trouble doing lookups against external LDAP directories, the `ldapsearch` utility can be very helpful. Try running `ldapsearch` with a wide search filter, such as `"cn=*"`, using the parameters you have set up in the Directory Assistance database (base DN, bind DN, and so forth). This should enable you to eliminate the LDAP component as a possible problem.

Also, make absolutely sure that your Directory Assistance configuration is correct. Remember that the default is to use SSL for the connection, so make sure that either SSL is running or the setting is changed in the Directory Assistance document. If you are running SSL, make sure that the Domino server and the LDAP server have a certificate in common.

### 5.3.4 Troubleshooting LDAP lookups against Domino Directory

Here are some problems that might arise when using the LDAP services of the Domino Directory:

- If you're using SSL to communicate between an LDAP client and the Domino server, make sure that the Domino server's certificate is one that is trusted by the client machine. If you have generated a certificate from an internal Certificate Authority, Domino or otherwise, you may need to connect to that server with a Web browser (Internet Explorer or Netscape Communicator, for instance), before connecting from an LDAP client (such as Outlook Express or Netscape Messenger). This will allow you to add the server's certificate as one that is trusted by the client.
- If you are trying to export the LDAP schema, and the operation is not successful, make sure that the `schema50.nsf` database is not open. This will prevent `load LDAP exportschema` from working, since it deletes the existing `schema50.nsf` and creates a new one each time.

- If LDAP clients are having trouble binding to the Domino Directory server, make sure that they are trying to bind with a name that is in the primary directory. Bind operations are not supported against secondary directories.

Make sure that the Domino Directory has a full text index. LDAP searches will use the (\$Users) view first, then try to use the full text index to get an answer to the query. Without an index, the search will be much slower.

Some browsers support using the `ldap://` URL to browse their directory. Netscape generally works better than Internet Explorer for this. For example, typing `ldap://balder.lotus.com:389` in the location bar and pressing Enter will bring up general information about the server. You can further browse by entering the DN of the entry that you want to examine in the URL string, like `ldap://balder.lotus.com:3890/cn=Jonathan Walkup/o=RedBook`. This will perform an anonymous bind and return the available information about the DN. For more information, see RFC 2255, The LDAP URL Format, in Appendix B, “LDAP and X.500 Standards” on page 159.

One tool we’ve found invaluable in helping us troubleshoot problems accessing the Domino LDAP server is an LDAP browser. You can download a nice Java-based browser from <http://www.iit.edu/~gawojar/ldap/>. Just the process of making this tool work with your Domino LDAP server can help you figure out configuration problems you may not have even discovered yet.

### 5.3.5 Troubleshooting authentication issues with external directories

When troubleshooting problems that Web clients are having authenticating against an external LDAP directory, it usually boils down to one of two things: a problem with the database ACL or a problem with the Directory Assistance configuration.

In database ACLs, make sure that the name you are using is the same as the distinguished name being returned by the LDAP server. An LDAP name like “`cn=John Wayne, ou=Heroes, o=westerns.com`” should be entered into an ACL as `John Wayne/Heroes/westerns.com`. If you have any doubt, use `ldapsearch` or a similar tool to retrieve the DN and make sure your ACL matches.

If you are using a group in the database ACL, make sure that the group name looks like the name of the group being expanded on the LDAP server. Most LDAP servers will present their groups in a hierarchy, even though Domino doesn’t assign groups to OUs or Os. You can simulate names like this, however, by entering groups with forward slashes to denote the hierarchy

presented by the LDAP server. For example, if the distinguished name of the group you are expanding is “cn=Outlaws, o=westerns.com”, you can enter that into the ACL as “Outlaws/westerns.com”, and it will match.

When troubleshooting Directory Assistance problems, first make sure that the Domino server is able to bind to the external LDAP server. If your LDAP server does not allow an anonymous bind, you will need to enter a distinguished name into the Directory Assistance configuration for that directory. Similarly, if your LDAP server requires SSL, you will need to enable that in the Directory Assistance configuration. SSL is the default setting for an external LDAP directory, so make sure to change it if you’re not using SSL.

Finally, make sure that the Domino Directory has the ability to read the required attributes from the external LDAP directory. The Domino server is going to present a request for multiple attributes that might match the name used to log in. Depending upon your selection of Web Authentication methods in the HTTP server section of your Domino server configuration, you may need to allow access on your LDAP server to more attributes. If you are unable to figure out why Web clients are unable to authenticate, you can use the WebAuth\_Verbose\_Trace notes.ini variable to help determine where the failure is occurring. Set the .ini parameter to “1”.

---

## **5.4 Guidelines and best practices for using the Domino Directory**

Here are some guidelines and best practices from our personal experience that should help you work through a directory implementation project.

### **5.4.1 Take an enterprise-wide approach**

The best first step in dealing with any directory issue is to step back and take a look at the big picture. Evaluate how this directory fits into the overall directory architecture of your environment. Look for other directory sources that could be integrated into this one, or that could use this directory’s data to be more effective.

Even if you are not implementing an x.500-based directory structure, it can be very useful to review some of the basic concepts of the x.500 architecture. The x.500 standard describes a separation of duties between directory users and servers, and possible combinations of directory servers that can be very useful when coming up with a simpler directory architecture. Furthermore, the richness of the x.500 directory model can be used as the basis for your own Notes hierarchy.

Whenever you're dealing with a multiple-directory scenario, one of the early challenges will be to define a unique key for each entry, so that you can synchronize entries. In fact, even if you're not dealing with multiple directories yet, it would be a good idea to include a unique key in your directory from the beginning. Most organizations use some sort of employee ID number for correlating person entries in different directories. If everyone in the organization uses internet mail, you could use SMTP addresses as unique keys. Try and find some common ground, or create it if it doesn't exist. When you later need to synchronize or merge directories it will be a lifesaver.

If you are designing a multiple-directory environment, be sure to lay out not only the different directories you will be integrating, but also the different tools and administrative processes that will keep them all in sync. It may be possible to combine two or more of these tools or processes to help reduce the number of possible points of failure. Using standards-based tools is almost always a better approach when working with multiple directories.

If you are going to be moving large amounts of entries or changes into your Domino Directory, be sure to schedule time for servers in your Domino environment to replicate the changes around and update indices to reflect the new contents. Make sure that your updates occur in a centralized location, to preserve data integrity, but allow time to replicate to your distant servers. Take advantage of Domino features such as field-level replication and selective replication when replicating directory data.

If you are going to be using the Domino Directory to store and manage directory entries that are not normally stored in it, make sure to design an interface for those entries within Notes. If possible, you might want to set up a separate directory for those non-Domino elements so you can customize more freely, without the risk of affecting normal Domino operations.

Determine how you are going to use LDAP in your environment. Notes clients can use LDAP accounts to look up names in an LDAP server. Domino servers can use LDAP with Directory Assistance to look up mail addresses and authenticate Web users. The Domino Directory can be exposed to external LDAP users, clients or servers for read/write access. Since LDAP is becoming the glue that holds disparate directory systems together, plan carefully how your entire environment will take advantage of it.

If you're migrating from another messaging system, or if you have entries in another directory, make use of the migration tools provided in the Domino Administrator, if possible. They will give you flexibility in creating new accounts that would be difficult to recreate programmatically.

#### **5.4.2 Managing multiple Domino directories**

One key point to remember when managing multiple Domino directories is that you can only use groups from the primary directory, and for Web users one secondary LDAP directory, to authorize users in an ACL. In a multi-domain environment, this is going to necessitate some sort of synchronization process for groups that are shared between the domains. This can be relatively easily accomplished with any number of methods (LotusScript, API, and so forth), but it needs to be taken into account up front when planning your infrastructure.

#### **5.4.3 Designing an efficient Domino Directory**

Keep the number of connection documents between servers to a minimum. Design your mail routing topology so that servers are grouped into Notes Named Networks for routing without connection documents, and implement a hub-and-spoke replication topology to keep connections simple and avoid nasty troubleshooting problems.

#### **5.4.4 Streamlining administrative processes**

Spend some time up front designing the user registration process. Define who will create new users. If you need custom data to be added for each new user, define the procedure for adding it to the directory. Work out naming standards, such as use of middle initials, the method for creating short names and internet addresses. If you're going to use alternate names, make sure your certifiers are set up with alternate naming information from the start.

Decide upon an administrative model. Will you make changes from a centralized location, or will you distribute that functionality to remote administrators? The centralized model is easier to control, but more difficult to make timely changes. Take advantage of the hierarchical structure of Notes to allow administrators in remote locations to register users in their Organizational Units, without giving them the ability to modify other users in the directory.

Leverage the Admin Process. Especially in a multi-server or multi-domain environment, it is well worth the extra planning up front to ensure a smooth deployment of the Admin Process. Use it to handle name changes and recertifications, instead of developing manual processes. Domino includes API access to the Admin Process; take advantage of it if you're using other directories to avoid duplication of efforts.

One of the biggest challenges for directory administrators is trying to maintain accurate data in the directory. Identify the administrators responsible for

maintenance of directory entries. Sit down and go over the person document and decide what fields you will populate with meaningful data. Don't use fields in the person document unless you have both a reason for doing so and a method for keeping them up-to-date. If you want to maintain personal contact information in the directory, consider giving end users access to update those elements.

Whenever possible, delegate group management to people who know the group's membership. Give the people who will be using the group responsibility for keeping it current. Use the [GroupModifier] role in the directory to grant this level of access.

Use the Domino Directory to maintain server settings. Server Configuration forms are a good way to keep .ini settings under control. Make a Configuration Document for every server in your organization, or every group of identical servers, and use a single global Configuration Document only for LDAP settings.

Use User Profiles to push configuration down to end users' desktops. Mobile directories, directory servers, and LDAP accounts are examples of good uses for setup profiles. Setup Profiles can also be used to push any field down to a location document, you just need to add a field to the profile document in the proper format. Append the fieldname that you want to set on the Location Document with "LocAll" on the Setup Profile document.

Make sure that you are protected in the unlikely event that your directory becomes corrupted. Make regular backups and store them in a safe place. If you have made customizations to the directory, make sure you have made those changes to a template, so you can easily reapply them if needed. Make sure that the ACL of the directory doesn't give any person or server more access than they need. Don't let any end users have the ability to delete documents unless absolutely necessary.

If you are in the process of migration from Release 4.x of Domino to Release 5, or if you are in a mixed environment, make sure that the design of directories on the R4 servers has been upgraded to the R5 design. The R5 design has been created to be fully backward compatible with R4 servers and clients, so you should upgrade as soon as possible. Also keep in mind that Directory Assistance requires that secondary directories have a Directory Profile, which is only present in the R5 design, to resolve ambiguous group names.



#### **5.4.5 Modifying the standard Domino Directory design**

Use the documented methods for modifying the Domino Directory. You will avoid the risk of modifications being overwritten by design changes and have the additional benefit of extending the design changes to LDAP users as well. Avoid customizing anything in the directory template that is purposefully hidden from view; you will find it difficult to get support when those changes cause problems. If, for some reason, you are required to change an element that is present in the default design, make sure that your changes are well documented and that you are aware of how those design changes can get overwritten by the design task or an individual with the standard template.

If you need to do significant modifications to the default design, consider separating that directory onto another server and preventing the design changes from flowing back into the rest of the environment, or perhaps even setting up a separate domain, if the directory entries can be segregated as well.

#### **5.4.6 Security considerations for a Domino Directory**

Since the Domino Directory by its nature contains a lot of information about people and resources, be careful to plan your security model beforehand. Fortunately, Domino has a very rich security model, and it is thoroughly implemented in the design of the Domino Directory. Here are some tips to help in planning your directory security model.

Use the roles that the Domino Directory template uses for its security. Do not give any end-users higher than Author access (Reader access is better, but will prevent them from updating groups and personal data, if you want to allow that). Give people responsible for creating and managing user IDs the [UserCreator] and [UserModifier] roles. If you are going to give end users the ability to edit groups they own, give them the [GroupModifier] role. Remember that Editor-level access is just about the same as Manager with respect to security.

Do not allow simple authentication (username and password) access to the Domino Directory from a Web browser. If you need to access or modify information from a Web browser that is in the Domino Directory, consider developing a separate front-end application to perform that operation. If you do allow simple authentication, certainly do not give users the UserModifier role, allowing them to edit and examine person documents in the directory.

Be especially aware of security issues if you use the Domino Directory to distribute Notes ID files. While this is useful if you need to distribute ID files to remote servers, be aware that any user who knows the password assigned to

the ID file can detach it and impersonate that user. Whenever possible, distribute ID files by a more secure mechanism.

If you deploy a Domino server that resides outside a corporate firewall, place the server in a separate Domino domain. You can still use the primary domain's directory for user authentication with Directory Assistance, although you might want to make a selective replica, with only username and password for example, to limit security risks.

Be aware of limitations with HTTP passwords. For example, there is no built-in mechanism for forcing HTTP password expiration or locking out HTTP passwords after a given number of invalid attempts. If you are managing sensitive data over the web, user certificates are generally a much better approach to take than simple authentication.

Be aware, also, of the features that Domino provides for checking Notes ID passwords. Password and Public Key checking can help secure your Domino environment from people who have stolen ID files and passwords.

#### **5.4.7 Leveraging Directory Catalog and Directory Assistance**

Directory Catalog and Directory Assistance allow you to extend your directory infrastructure to include multiple different directories. Take advantage of that fact. Look for directories in your enterprise that might be helpful for mail addressing or authentication. Most directory vendors support LDAP now, and even those that don't can probably be integrated into your directory structure, with some creativity. If the directory is not LDAP-accessible, but can be accessed via ODBC, for instance, consider creating a secondary Domino Directory and populating it via DECS.

Another possibility is using Directory Assistance to point to extranet or internet LDAP servers. Perhaps you're always dealing with end users who have trouble getting the address right for an important customer. If their directory can be exposed via LDAP, you could put an entry into Directory Assistance for their directory, allowing your end users to address mail without having to worry about getting the address exactly right. Perhaps you want to let your end users address mail using one of the large internet LDAP directories, such as Bigfoot or 411. You can set up an entry in Directory Assistance for that as well.

The bottom line: be creative. Look for opportunities to use the functionality that comes with the Domino Directory server.

#### 5.4.8 Leveraging the inherent benefits of the .nsf format

One of the biggest benefits of the Domino Directory, and certainly one of the most overlooked, is that it is based upon Domino database technology. This technology enables many possibilities; here are some examples:

- Take advantage of Domino's replication abilities. If you would like to provide multiple directory servers in geographically disperse areas, all with the same directory information, you can use replication to keep the information in sync. Additionally, if you'd like to provide a high-availability solution, you could use clustered servers to provide 100% uptime.
- Use Domino Designer to add application logic to your directory data. If you need to manipulate data in your directory as it is entered via LDAP, you can use the Domino Designer IDE to add validation formulas and translation formulas to the directory entries. If you need to add additional attributes to the LDAP object classes, you can use the Domino Designer as well.
- Use selective replication to present varied views of your directory. If you need to have separate data sets in your directory, for internal vs. external use, for example, you can use selective replication or replication formulas to create a partial replica of the dataset.
- Use the Domino security model to keep information in your directory secure. If there is sensitive information that you need to secure in your directory, take advantage of database-, document- or field-level encryption to keep it secure.

#### 5.4.9 Enforcing schema checking

The default behavior of the Domino LDAP service is to not enforce schema checking when new entries are added to the directory. Turning on schema checking will require that all entries added to the directory conform to the existing schema. If they use object classes or have attributes that are not defined in the Domino Directory's schema, the operation will fail. To enable this feature, you need to add the setting `LDAP_Enforce_Schema=1` to your `notes.ini` file and restart the server.

**Note:** For a discussion of *schema*, see Appendix H, "Directory Schema" on page 253.

If you have schema checking turned on, and you try to add an entry to the directory that doesn't have an object class that is defined in the LDAP schema, your LDAP client will get a failure with error code 65: objectclass not defined in schema. Follow the instructions in 4.2.4, "Modifying the LDAP Schema" on page 53 to add the object class to the Domino schema.



---

## Chapter 6. Using Domino Directory services

In this chapter, we show some methods for accessing Domino directories, using a variety of clients, standards-based tools, and application development programs.

---

### 6.1 Client access to Domino Directory services

The purpose of this section is to provide a high level review of using the Domino Directory services from a variety of clients, focused first on the Notes client, then highlighting the differences for other clients using Internet protocols, especially the Web protocols.

#### 6.1.1 Domino Directory services for the Notes client

The initial perspective, and seemingly the most straightforward, is that of a Notes client. The landscape from this perspective can be either boringly straightforward, involving just the Notes client and the primary directory, or very rich and sophisticated. The latter case arises as infrastructure planners augment the primary directory using the full range of Domino Directory services. These include a server-based Directory Catalog if their environment includes other organizations that have Domino directories that can be concatenated. They can create a Mobile Directory Catalog to give users the ability to look up names when disconnected, or to save server resources by offloading type-ahead functions onto the client. And they can leverage the use of Directory Assistance to access additional Domino directories that may not be appropriate to include in the Directory Catalog, or to extend address lookups to external LDAP directories, intranet-, extranet-, or Internet-based.

For the rest of this section we'll review how this rich tapestry impacts the user for three primary functions -- mail addressing and verification, authentication, and authorization.

##### 6.1.1.1 Mail addressing and verification

There are two major ways that users can address new messages: Either through use of the address picker dialogue, or by manually typing in the addressee's name. If the user decides to leverage the address picker dialogue, he further has the choice of using either Domino Directory services or the client's native LDAP capabilities to access the target directory or directories. However, since the vast majority of e-mail users are not employed to be protocol experts, the details of which protocol is being used is hidden. Likewise, the configuration of Directory Catalogs and Directory Assistance is not a user responsibility.

### **Address Picker**

The address picker dialogue is accessed via the Address button on the action bar, or the Address selection on the Action menu. See Figure 14.

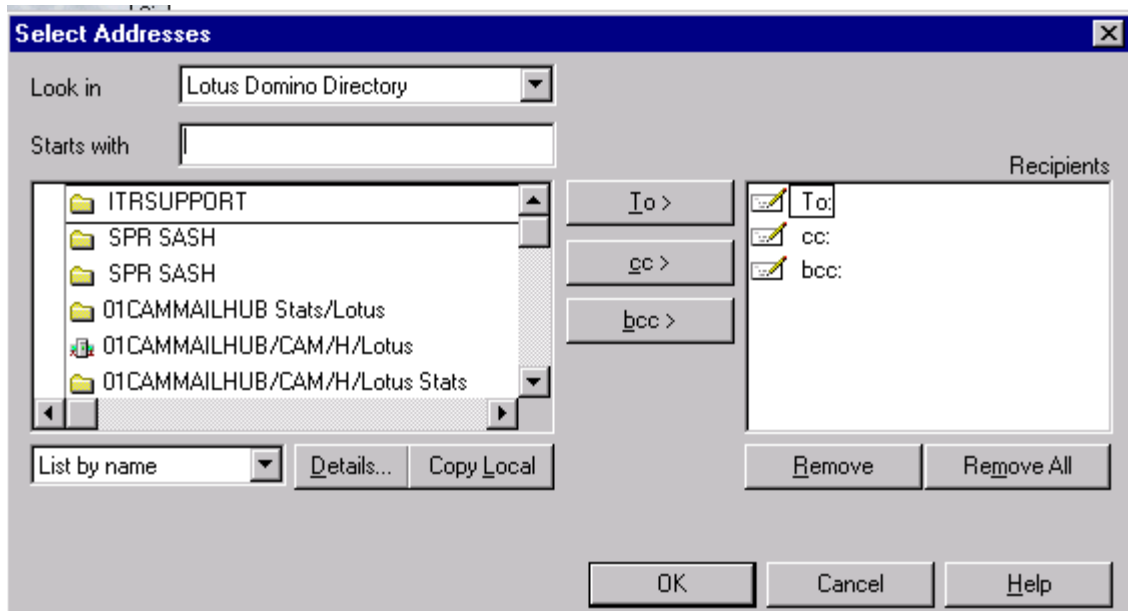


Figure 14. Address picker dialogue

The “Look in” drop-down list displays all directories that are available for users to locate addresses. When using Domino Directory services, these directories include directories held locally on the client, the primary directory on either the mail or directory server (depending on the location document settings; the default is to use the mail server), the Directory Catalog, and all databases configured in Directory Assistance. In Figure 15 on page 91, access to all of the IBM directories displayed is provided by Directory Assistance on a directory server.

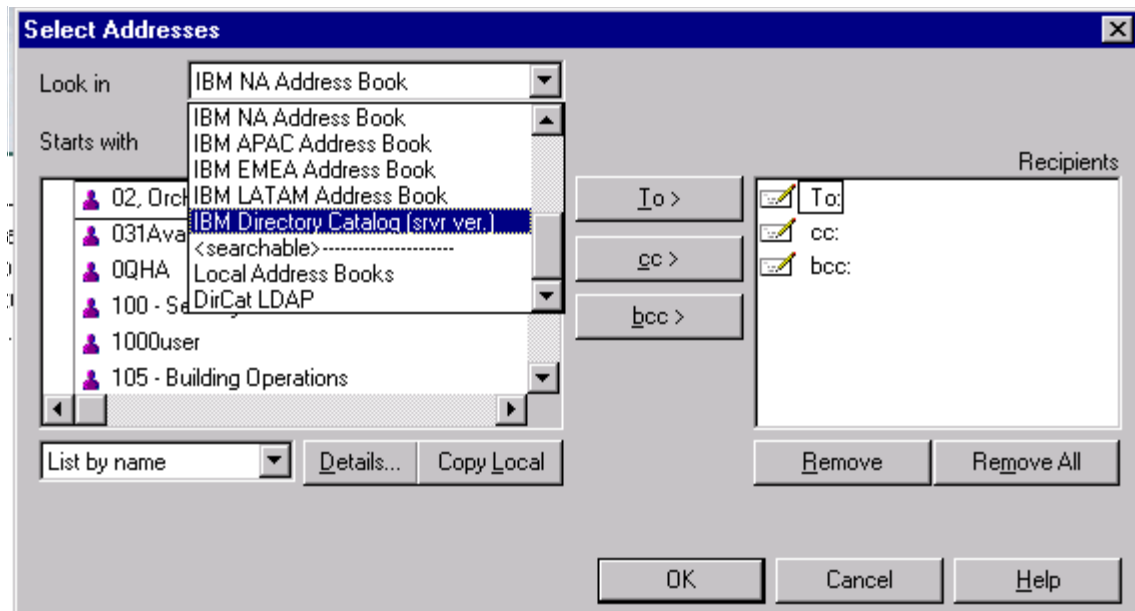


Figure 15. Display of available directories

As the user begins to enter a name in the “Starts with” field, Domino Directory services jumps to the first entry in the selected directory matching those letters in the names list window. See Figure 16 on page 92.

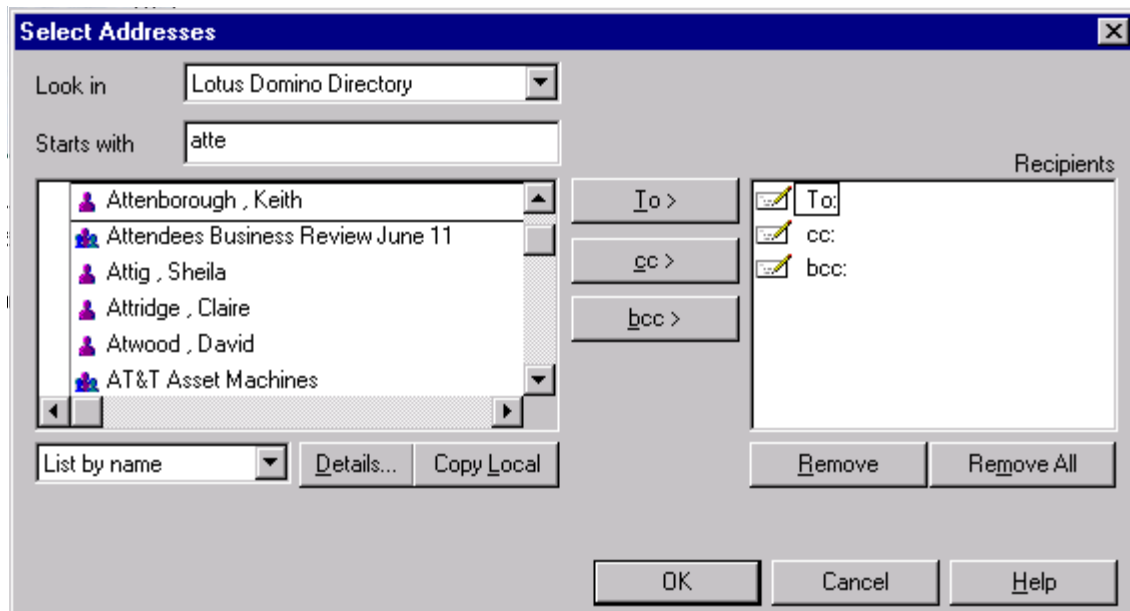


Figure 16. Quick find locating an entry

When the proper name is found, the user either clicks **To**, **cc** or **bcc**, or drops-and-drags to populate the appropriate addressing field in the memo.

Native client LDAP searches can also be accomplished via the address picker. These require a preliminary step, which is to set up an Account document in the individual local address book. These Account documents can be created either by the user or by the administrator and pushed to the user via profiles documents (these procedures are covered in the administration documentation). The directories that are accessible via these procedures also appear in the "Look in" drop-down list, but are placed below the <searchable>----- bar, as seen in Figure 17 on page 93. Note that address books on the client are available by default and do not require an Account document.



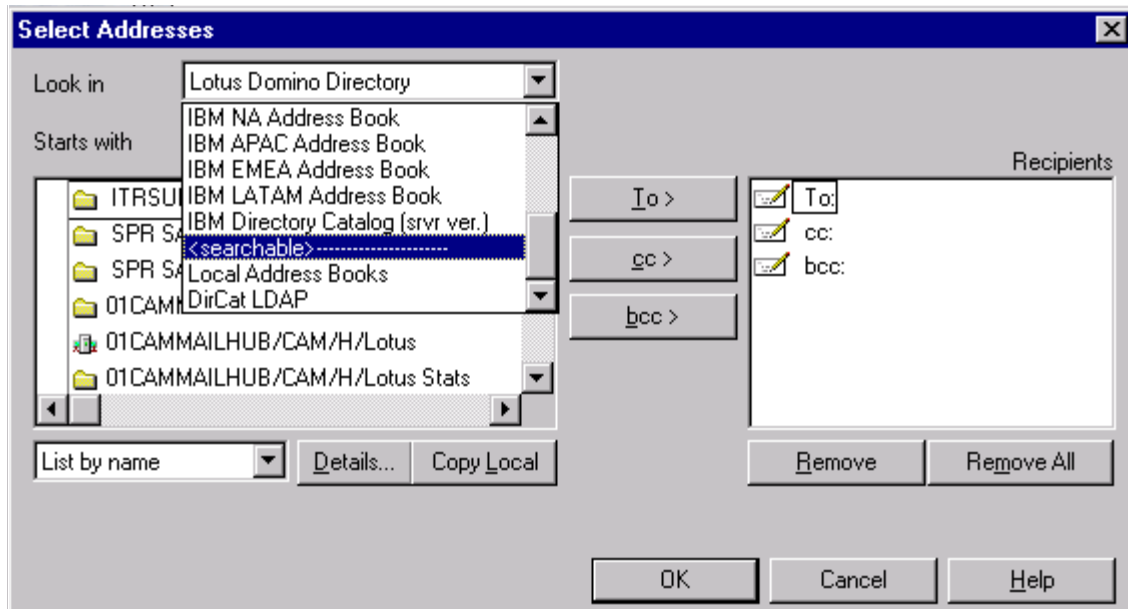


Figure 17. Directories displayed below the searchable line

Selection of an LDAP-accessible directory causes the address picker dialogue to change appearance, reflecting the fact that the LDAP protocol currently supports a search-and-return process, rather than providing a scrollable list.

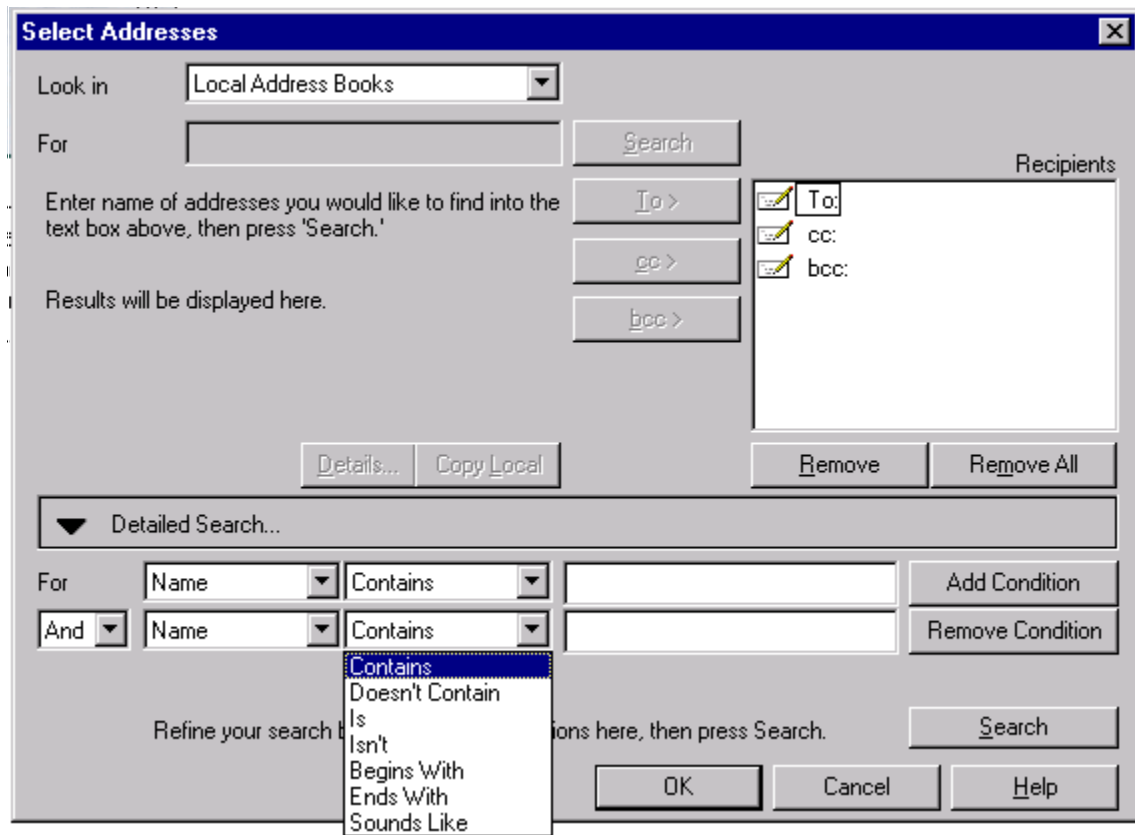


Figure 18. Search panel provides multiple options

The “For” entry box just below the “Look in” field is normally available and supports simple searches. The box greys out when the Detailed Search section of the dialogue is opened, and as is shown in Figure 18, extensive search options are available. The selected LDAP directories can be local, or can be hosted via enterprise intranets or the Internet. Local address books include any address book listed in the “Local address books” field of the Mail and News panel of the User Preferences dialogue. Note that the address picker dialogue does not screen out duplicate names, so if users have multiple Directory Catalogs with overlapping entries on their clients, they are apt to see multiple entries for a given search. The results panel displays both the user name and the e-mail address to assist in selecting the correct instance. Once selected, the To, cc and bcc fields are populated in the same manner as with entries found via Domino Directory services.

### **Manual Entry**

In addition to the address picker dialogue, users can manually enter names in the addressing fields of a message. To assist users in quickly locating a name, the Notes client provides both Type Ahead and Type Down services. These were discussed in Chapter 2, “Domino Directory services” on page 5, but deserve a brief review here.

Type Ahead and Type Down are client services designed to help users quickly enter a target address, if that address is located in either the local address books, the primary Domino Directory, the server Directory Catalog or secondary Domino Directories configured in Directory Assistance. It does not access LDAP directories configured in Directory Assistance in order to preserve reasonable response times. Type Ahead parameters are location sensitive and therefore are configured on the Location document. Parameters include whether Type Ahead is enabled, whether it will only search local address books or extend its search to the server, and whether it is activated with each character or only when a delimiter (such as a comma) is entered. Type Down is a complementary service that permits a user to scroll through the entries contiguous to the initial *hit* using the up and down arrow keys on the keyboard.

While Type Ahead has an ambiguous name resolution capability, it is not the same procedure and not as robust as the full address verification process discussed next. The Type Ahead ambiguous name resolution capability, like Type Ahead itself, is designed as a fast responding *helper* and in the vast majority of instances provides significant value to the end user by quickly locating the correct mail address. However, definitive results are obtained via the more thorough address verification process.

Manual entry also permits the user to enter names not found in any directory, a capability primarily used for Internet addresses. These manual entries will be successfully used to route messages as long as the appropriate routing configuration documents have been placed in the primary Domino Directory. See the Administration Guide for more details.

### **Address Verification**

This was also covered in Chapter 2, “Domino Directory services” on page 5, but again deserves a brief review here. The Notes/Domino address verification processes are used by the mail router to ensure mail can be delivered to an unambiguous address. In performing this function, the mail router code will search the primary Domino Directory, the Directory Catalog, all secondary Domino Directories enabled in Directory Assistance and then all LDAP-configured directories enabled in Directory Assistance.

The identical address verification processes are invoked from the client, either by pressing the F9 key or by initiating the Mail Send function from the action bar or menu, and a second time by the mail router as it processes the message. One clear indication of this is the use of an identical ambiguous names dialogue box with an option button to Cancel Sending, even when F9 has been used to invoke the verification process.

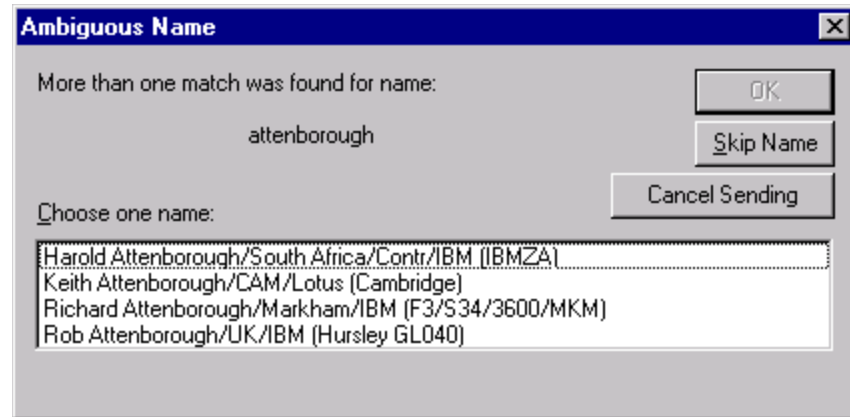


Figure 19. Ambiguous Name dialogue resulting from pressing F9

Address verification is sensitive to one configuration parameter. For the user, this parameter, “Recipient name look up”, is considered location sensitive and is found on the Mail tab of the location document. The choices are either “Stop after first match” or “Exhaustively check all address books”. The “Stop after first match” setting will cause the ambiguous name process to only examine other names in the Directory in which the first match was found, while the alternate setting will continue to look in additional configured directories. For the router, the same parameter is called Exhaustive Lookup and is set on the Basics subtab of the Router/SMTP tab of the Configurations Settings document. It can either be Enabled or Disabled and by default is Disabled. Enabling the parameter on the server can adversely impact mail delivery performance and increase the size of mail queues.

#### 6.1.1.2 Authentication

Authentication is the process of validating the identity of a specific individual entity, most frequently a person, though it can also be an application or server. The basic functionality uniquely identifies users and programs, verifies these identities and assures individual accountability.

A critical point originally made in Chapter 2, “Domino Directory services” on page 5 that needs to be reiterated is that authentication is the process of

validating an individual identity. These individual identities may be combined into groups or mail lists or other aggregations, but only the individuals are authenticated. This comes to bear later when the discussion turns to authorization.

The Notes authentication process is based on certificates and the use of public/private key cryptographic techniques. The certificates and public keys used by the Notes authentication process are created and stored in the Domino Directory for use by users and Domino applications. The specific process used by Domino is outlined in 2.6, "Authentication and security services" on page 21 and in other publications, so it will not be repeated here.

From a user perspective, the authentication process is as straightforward as correctly entering their Notes password when prompted. The extent and complexity required to positively verify that user's identity, based on a two-factor system (the user's possession of the Notes .id file and knowledge of the correct password) and an extensive public key infrastructure, is almost totally transparent to the individual at the keyboard, to whom it appears no more difficult than logging on to the Web.

From an administrator's and infrastructure architect's viewpoint, it is important to know that Notes users are only authenticated based on entries in the primary Domino Directory, and not via Directory Catalog or Directory Assistance.

#### **6.1.1.3 Authorization**

Authorization is the process of determining what databases, applications and services an authenticated individual has access to or can invoke. Said another way, it is the Domino environment saying, "Okay, now that I know who you are, I can figure out what you're allowed to do". Clearly the critical and necessary first step is to establish the authenticated identity as described in the previous section.

In the authorization process, it is the value that the Directory returns as the authenticated identity that is used to determine the rights of the user. This value is most frequently the Distinguished Name (first entry in the Full Name field) included as part of the person document that held the credentials used to authenticate the individual entity. This authenticated identity is presented to the access control process protecting the requested information or service. For Domino databases and applications, this is the access control list of ACL. If the authenticated identity is included in the ACL, then the appropriate level of privilege is determined and access granted.

Clearly a critical juncture in the process is ensuring that the authenticated identity matches the ACL entry. For a Notes client, the most secure method of doing this is to enter the fully distinguished name in its hierarchical format (Keith Attenborough/CAM/Lotus) in the ACL. However, if the user name of the person is in the same hierarchical organization as the Domino server on which the database is stored, you can enter only the common name portion (Keith Attenborough). If the user is in a different hierarchical organization than the server, the fully distinguished name must be used.

Group names can be used during the authorization process to grant access to data and services. Since only individual entities, not groups, can be authenticated, the use of groups in ACLs is a two-step process. The first step is for the authorization process to determine to which groups an individual authenticated entity belongs. In Domino, this is done as soon as the authentication process completes. Essentially, once an entity's identity is authenticated, a search is done to determine which groups contain that authenticated name and a list is built and cached with those group names. As described in Chapter 2, "Domino Directory services" on page 5, this process is called *group expansion*. When the individual attempts to access an application or service, the authorization process presents the list of names that includes both the individual and group names associated with the authenticated identity. For Notes clients, group expansion only occurs in the primary Domino Directory.

The second step is to enter the group name in the ACL. As with individual identities, the group name must be entered in the ACL exactly as it will be presented by the authorization process. In Domino, group names are non-hierarchical.

## **6.1.2 Differences for Internet clients**

There are perhaps three significant differences between Notes clients and other Internet clients (Web browsers, Internet mail clients, applications) that warrant highlighting in this chapter. These differences fall into the same three categories used to describe the Notes client capabilities -- Mail Addressing, Authentication and Authorization.

### **6.1.2.1 Mail Addressing**

The key differences here are related to the User Interface, or UI. Each vendor's client has its own unique way of establishing the needed configuration to access directory services, and how mail address information is presented. As a class, however, these differing UIs are sufficiently similar so that most users can move from one to another without a great deal of difficulty. They should simply be led to expect differences and to have some

short orientation period before they are fully comfortable in the alternative vendor's product.

Regardless of UI, however, mail address lookup by Internet clients using LDAP is fully supported in the Domino Directory architecture, and these clients can locate addresses (to the extent they are authorized access) in the primary Domino Directory, the server based Directory Catalog, secondary Domino Directories configured in Directory Assistance and third-party LDAP directories configured in Directory Assistance.

#### **6.1.2.2 Authentication**

The relevant information here is that Domino will support authentication of Internet clients whose credentials are held in the primary Domino Directory. More importantly, the Domino Web server will also authenticate Web clients whose credentials are stored in secondary directories, including LDAP directories, configured in Directory Assistance with a rule marked as "Trusted for Credentials". These credentials can be either user id and password or X.509v3 certificates. Further, authentication of Web clients can be facilitated by placing the credentials in a secondary Domino Directory, then aggregating that Directory into a Directory Catalog. In this case, the authentication process will use the credential stored in the Directory Catalog (ahead of Directory Assistance in the search order), as long as the source secondary Directory is in Directory Assistance and Trusted for Credentials, without doing a lookup in the source directory itself.

#### **6.1.2.3 Authorization**

The key difference between the Notes client and Internet clients during the authorization process is that while group expansion will only occur in the primary directory for Notes clients, the administrator can additionally designate one third-party LDAP directory configured in Directory Assistance to support group expansion for Web clients.

In addition to this difference, there is also a similarity that can cause confusion. As with the Notes client, the Web client authorization process begins with the presentation of an authenticated Distinguished Name to an access control point, most frequently the ACL of a database. What needs to be remembered is that this presented DN will be the value returned by the authenticating directory. If that authenticating directory is a third-party LDAP directory, this DN may have a significantly different format than a Domino DN. For example, it may include one or more `dc=xxx` values. In order for authorization to be successful, the entry in the ACL must match this value.

Presently, the ACL editor does not allow lookups against third-party LDAP directories. Therefore, DNs with unique values must be hand-entered into the ACL for authorization to succeed. While laborious, this will work. Two additional notes: (1) the canonical name form must be entered, that is, the attribute titles must be included (cn=x/ou=y/etc.); and (2) if the DN is returned in LDAP format, that is with commas as separators, the commas must be entered in the ACL as forward slashes. For example, cn=Keith A,ou=cam,o=lotus,dc=kayak would be entered as cn=Keith A/ou=cam/o=lotus/dc=kayak.

---

## 6.2 Using standards-based tools for directory manipulation

The Internet community has been steadily developing various standards to make manipulation of directories much easier for administrators. In this section we investigate some of those standards and some tools that can be used to add and modify data in a Domino Directory.

### 6.2.1 LDIF

LDAP Data Interchange Format (LDIF) is the standard method for moving data between different directories. You will inevitably come across situations where you will need to import or export an LDIF file. Thankfully, Lotus has provided tools with Domino that make that import/export task very simple.

#### 6.2.1.1 Importing LDIF data from other directories

The Domino Directory can use LDIF to move data in from other directories and export directory information that can be used by other directory servers. Since importing LDIF data is, in a sense, similar to registering new users in the Domino system, the tool that is used to perform those operations is the Domino administrator.

To import an LDIF file into the Domino Directory, open the Domino Administrator client, make sure you are working with the proper server, and click on the **People & Groups** tab. Now expand the People section under Tools, and click **Register...** You will be prompted for your certifier location and password (only the password after the first time). This is necessary if you want to create the Notes ID, and you must enter it regardless. Note that the choice of certifier that you use will also define where in your Notes hierarchy the person documents will be created. If you need to import LDIF entries into multiple different OUs in Domino, you will need to create separate LDIF files for each OU. Once the Register Person dialog box has come up, click **Migrate People...** This will bring up the Migrate People dialog box. From the Foreign directory source drop-down menu, choose **LDIF Entries**. Now choose the



LDIF file that you want to import. Once you've chosen it, you should see a screen as shown in Figure 20.

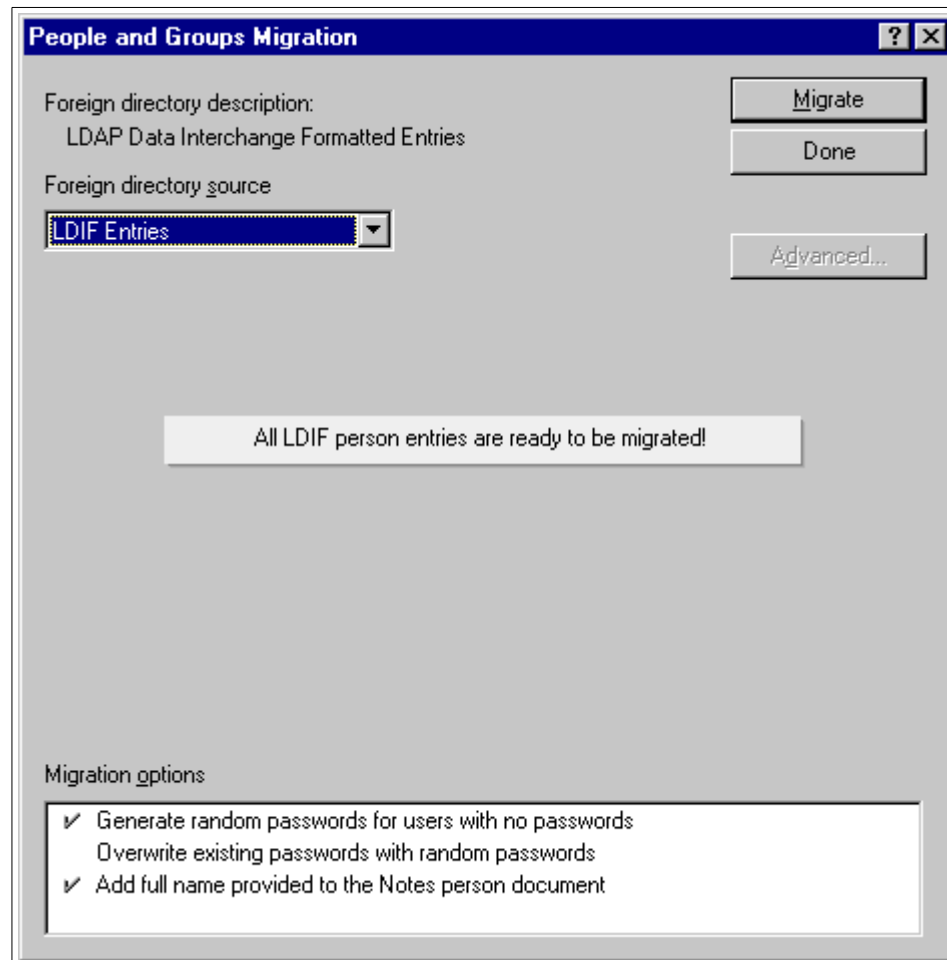


Figure 20. LDIF import dialog

Once you have selected the file, you have three migration options at the bottom of the dialog. Options 1 and 2, "Generate random passwords for users with no passwords" and "Overwrite existing passwords with random passwords" cannot be used together. The final option, "Add full name provided to the Notes person document," will add the distinguished name of the imported person into the FullName field of the person document. When you've chosen any options you wish, click **Migrate**. The new users will be added to the registration queue.

If you are adding these users into your Domino Directory as regular Notes users, with Notes clients and mail files, you can now click **Register All**. If, however, you are adding these users as Internet mail clients, or as directory entries only, you will need to modify the default settings for the registration process. First, select all of the entries in the registration queue that you just imported from the LDIF file. If you want to make them Internet mail-only users (for standard POP3 or IMAP use), check the Advanced check box and click **ID Info**. Now make sure that both options for ID storage are unchecked. Since POP3/IMAP users don't use the Notes client, they won't need an ID file. If you don't even want to create mail files for these users, click **Mail** and change their mail system to None. Now you can click **Register All** to add them to the directory.

Please note that the LDIF import tool only imports the following types of entries:

- person
- organizationalperson
- inetorgperson
- internetperson
- residentialperson

All other entries will be discarded.

#### 6.2.1.2 Exporting LDAP data for use with other directories

The LDAPSearch utility can be used to create an LDIF file that can be imported into other directory servers. An example of such a command would be:

```
ldapsearch -h balder.lotus.com -p 3890 -b "o=redbook" -D "cn=Jonathan Walkup,o=RedBook" -w password -L "cn=*" 
```

This will generate output like the following:

```
dn: CN=Alfred LaRue,O=RedBook
cn: Alfred LaRue
shortname: ALaRue
uid: ALaRue
mail: Alfred_LaRue/RedBook@lotus.com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: dominoPerson
```

```
creatorname: CN=Alfred LaRue,O=RedBook
mailsystem: 100
messagestorage: 1
encryptincomingmail: 0
checkpassword: 0
availablefordirsync: 1
passwordchangeinterval: 0
passwordgraceperiod: 0
givenname: Alfred
sn: LaRue
userpassword: (F649A890EC6499FF6EA51F26B9BB73DD)
...
```

### 6.2.2 LDAPsearch

One useful tool that Lotus provides with Domino Release 5 is LDAPSearch. This tool allows you to test LDAP connectivity from the command line. Here are some examples of LDAPSearch commands. See Appendix D, “Syntax of LDAPSearch command” on page 201 for the full syntax:

Searching for all entries with the last name of “Autry” like so:

```
ldapsearch -h balder.lotus.com -p 3890 "sn=Autry"
```

produces the following output:

```
CN=Gene Autry,O=RedBook
cn=Gene Autry
shortname=GAutry
uid=GAutry
mail=Gene_Autry/RedBook@lotus.com
objectclass=top
objectclass=person
objectclass=organizationalPerson
objectclass=inetOrgPerson
objectclass=dominoPerson
givenname=Gene
sn=Autry
```

If you do an authenticated bind to the LDAP directory, and you have the appropriate access, you will see more information, as in this example:

```
ldapsearch -h balder.lotus.com -p 3890 -D "cn=Jonathan Walkup,o=RedBook" -w
password "sn=Autry"
```

which produces this output:

```
CN=Gene Autry,O=RedBook
cn=Gene Autry
```

```
shortname=GAutry
uid=GAutry
mail=Gene_Autry/RedBook@lotus.com
objectclass=top
objectclass=person
objectclass=organizationalPerson
objectclass=inetOrgPerson
objectclass=dominoPerson
creatortname=CN=Gene Autry,O=RedBook
mailsystem=100
messagelstorage=1
encryptincomingmail=0
checkpassword=0
availablefordirsync=1
passwordchangeinterval=0
passwordgraceperiod=0
givenname=Gene
sn=Autry
userpassword= (F649A890EC6499FF6EA51F26B9BB73DD)
```

### 6.2.3 Other command line tools

Included in the Netscape/iPlanet LDAP SDK are three command line tools that can be used for basic operations against any LDAP server. Their command syntax is very similar to LDAPSearch. For more information, consult the Netscape LDAP SDK.

#### 6.2.3.1 LDAPCMP

LDAPCMP compares attribute values for a given bind in two separate directories and determines if they are equivalent. If you have two separate LDAP servers and want to make sure that an entry is the same on both servers, you can use this command.

#### 6.2.3.2 LDAPDELETE

LDAPDELETE deletes entries from the directory server. This can be handy if you want to write a batch file or script to delete entries from an LDAP server.

#### 6.2.3.3 LDAPMODIFY

LDAPMODIFY will modify attributes of entries in the directory server. Again, it can be useful in making mass changes to a directory via LDAP.

### 6.2.4 LDAPSsync

LDAPSsync is a solution developed by the EMEA (Europe, Middle East, Africa) group of Lotus Professional Services.

It is a set of server utilities that provide a means to synchronize Notes Databases with external directories supporting the LDAP protocol.

It essentially provides the following features:

- Parameters extraction

An administrator may inspect the various attributes that are stored in a remote Domino or non-Domino Directory. This is especially useful when trying to define what may be synchronized.

- Notes to LDAP synchronization

This is a means to automatically synchronize the content of a Notes document field with the corresponding attribute in the directory. With this feature, attributes stored in an external LDAP directory may be imported into a Notes database.

- Synchronize two Notes databases

This tool may be used to replicate information between two Notes databases that do not share the same design, so they cannot be replicated with standard Domino replication.

LDAPSync uses a dedicated Domino database containing configuration documents as well as execution logs.

The toolkit is currently available for the NT platform. It uses Domino Server Version 4.5 or later.

Please contact Lotus Professional Services for more information or for customer references.

---

## **6.3 Using application development tools with Domino Directory services**

If you wish to access the data that is in a Domino Directory with a program, there are many possible ways to accomplish that goal. In this section, we'll show examples using the Notes API, the LDAP API, and JNDI. Additionally, we'll talk about other APIs and programming tools that you can use with the Domino Directory.

### **6.3.1 Using the Notes API**

The Notes API is a C language programming interface for Lotus Domino. The API can be used to access all of the different elements of Notes, including the Domino Directory. You can find the latest version of the Notes API at

<http://www.lotus.com/developer>.

With the Notes API, you have low-level access to the internals of any Domino database. The Domino Directory is no exception. You can use the API to add both documents and design elements, and in so doing add entries to the directory, or extend the directory schema.

See Appendix F, “Sample code using the Notes API” on page 205 for an example of Notes API code to add a subform to the directory and add a field to the subform, thereby extending the domino schema. It uses version 5.0.4 of the Notes API. This code performs the same operation that we did manually in 4.2.4.1, “Adding new attributes to the existing schema” on page 53. Using the Notes API to generate design elements is quite complicated. We recommend that you use the Domino Designer UI whenever possible.

### 6.3.2 Using the LDAP C API

You can use the LDAP API to write programs that bind to, search, add, update, compare, and delete from a Domino server that is running the LDAP service. Here’s an example of code to bind to an LDAP directory, search for a given user, and delete him:

```
/* example program to demonstrate the LDAP C API */

#include <stdio.h>
#include <ldap.h>

#define SEARCHBASE "o=RedBook"

int main()
{
    LDAP      *ldapconn;
    LDAPMessage*results;
    LDAPMessage*entry;
    char      *username="cn=Jonathan Walkup,o=RedBook";
    char      *password="password";
    char      *dn;

    /*open your LDAP connection */
    if ((ldapconn=ldap_open("balder.lotus.com",3890))==NULL)
        exit(1);

    /* perform a simple bind against the LDAP server */
    if (ldap_simple_bind_s(ldapconn,username,password) !=LDAP_SUCCESS) {
        ldap_perror(ldapconn,"ldap_simple_bind_s");
        exit(1);
    }
}
```

```

/* find the users that match the given searchbase */
if (ldap_search_s(ldapconn,SEARCHBASE,LDAP_SCOPE_SUBTREE,"cn=Bill
Gates",NULL,0,&results) !=LDAP_SUCCESS) {
    ldap_perror(ldapconn,"ldap_search_s");
    exit(1);
}

for (entry=ldap_first_entry(ldapconn,results); entry !=NULL;
    entry=ldap_next_entry(ldapconn,entry))
    if ((dn=ldap_get_dn(ldapconn,entry)) != NULL) {
        if (ldap_delete_s(ldapconn,dn) !=LDAP_SUCCESS) {
            ldap_perror(ldapconn,"ldap_delete_s");
            exit(1);
        }
        ldap_memfree(dn);
    }

ldap_msgfree(results);
return(0);
}

```

### 6.3.3 Other programming tools

Since the Domino Directory can be used as a standards-based LDAP directory, there are numerous programming interfaces that you can use to access its entries. Here are a few programming interfaces you might want to use in your directory infrastructure.

#### 6.3.3.1 JNDI

The Java Naming and Directory Interface classes can be used to write Java applets or applications that use LDAP to access or modify an existing directory. Since Java is cross-platform and supported in most Web browsers, it is a logical choice for self-service Web applications, for example. For more information on JNDI, consult Sun's Web site at

<http://java.sun.com/products/jndi>.

#### 6.3.3.2 DSAPI

Although it is not an API to a directory per se, DSAPI can be used to rewrite the authentication process for a Domino Web server, so it can extend your authentication method to any external data source. If you're using Domino as a Web server, you could rewrite the authentication process to use an external database as a directory source, or to pull custom attributes from an LDAP directory, for example.

#### **6.3.3.3 ADSI**

Active Directory Service Interface is the Microsoft-published API for Active Directory. While ADSI provides a direct interface into the internals of Active Directory, it is primarily a proprietary interface that does not extend beyond Active Directory to provide a tool set for interacting in a multi directory environment. Lotus is using LDAP as its interface mechanism to Active Directory as a strategy to enable using applications to be directory independent.

#### **6.3.3.4 PerLDAP**

PerLDAP is an open-source project working on LDAP extensions to the Perl programming language. As Perl is ideally suited to handling text strings, it is used commonly to process or generate Web pages, and is very commonly used in a Web server environment. Adding LDAP functionality to Perl allows Web page programmers to use an LDAP server to determine how to manipulate an HTML page, for example.

Perl is also used extensively by UNIX system administrators to automate common tasks. If you are trying to synchronize UNIX username and password files with a Domino Directory, PerLDAP may provide a programming interface that your UNIX system administrators and developers are more comfortable with than C or Java.

#### **6.3.3.5 Isxldap**

The LotusScript language, used internally by Notes and Domino, as well as other Lotus products, has the ability to be extended using LotusScript Extensions (LSX) files. One LSX file has been developed by Robein Shi of Lotus and submitted to the Iris Sandbox area on <http://www.notes.net>. If you are looking to programmatically access an LDAP directory from LotusScript, in an agent for example, check it out.



---

## Chapter 7. Directory integration

As businesses incorporate Internet technologies into their core business processes, they start to achieve real business value. Today, companies large and small are using the Web to communicate with their partners, to connect with their back-end data systems, and to conduct electronic commerce.

The Application Framework for e-business network infrastructure provides a platform for the entire e-business environment and includes TCP/IP and network services, security services, mobile services, client management services, and file and print services. Supported standards include LDAP, TCP/IP, Secure Socket Layer (SSL), and X.509v3 certificates among others. Central to all of these are directory services and directory standards, acting as the keystone of the framework's network infrastructure.

Now that directory standards are emerging, the opportunity exists for enterprises to simplify their directory issues. If an enterprise is able to store directory information once, the total cost of maintaining this information can be significantly reduced. However, in some enterprises, specific application requirements and migration costs may dictate that multiple directories continue to be maintained. In these environments, directory standards enable central administration and reduce the cost and complexity of synchronization. Given the distributed and heterogeneous nature of network computing, directory architecture is defined to satisfy the following requirements:

- Directory services must be based on existing or emerging standards when available.
- Access to directory services must be available from a variety of client platforms through a standardized API and standard Web browsers.
- A common schema for universal objects, such as users and groups, must be standardized to eliminate application dependence on specific directory implementations.
- Information maintained in existing directories must be accessible without requiring a complete migration to a new directory.
- Central administration of common objects stored in multiple different directories must be provided.
- Directory services must be secure. Access to information in the directory must be guaranteed for authorized users and denied for non-authorized users. In addition, the directory architecture must allow for storage and management of security information.
- Directory services must be available when needed by applications.

---

## 7.1 Overview

Domino R5 Directory is designed to serve as an integration point for directory synchronization, administration, and authentication in a heterogeneous networking environment with other directories. Enterprise directory features in Domino include:

- LDAP Version 3 protocol support in both the client and the server
- Access for lookup and application support for non-Notes clients (for example, Web browsers) and servers (for example, NDS or iPlanet) via standard protocols and third party LDAP toolsets, as well as the well documented Domino/Notes API toolset
- An extensible directory schema for customizing the directory to meet specific business requirements
- Multi-master replication
- Rule-based domain relationship, for faster lookups across large namespaces
- Support for LDAP/X.500 naming conventions, including hierarchical naming and extensible attributes for maximum flexibility in configuring a corporate directory

The purpose of this chapter is to provide examples of how Domino Directory services integrate into today's complex multi-directory environment. As you will see, in almost all instances Lotus has focused on leveraging industry standards, primarily LDAP, to provide a consistent experience regardless of product mix, assuming all products are equally standard-compatible. We firmly believe this approach provides the maximum benefit to our customers.

---

## 7.2 Integrating with other directories and applications

In this section, we show some of the many ways that the Domino Directory can be integrated with products from other vendors. We cover products from IBM and Lotus, Microsoft, iPlanet (the Sun/Netscape alliance), Novell, and others.

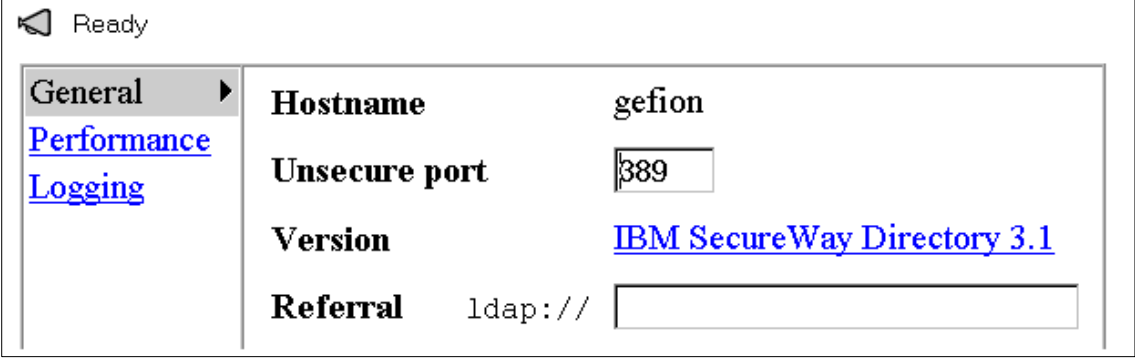
### 7.2.1 IBM and Lotus products

IBM markets an LDAP directory called SecureWay. It comes bundled with other IBM products and platforms. You can integrate SecureWay and Domino using LDAP as the common protocol. Additionally, you may want to integrate the WebSphere application server with Domino, or directories used by IBM

operating systems (OS/390 or OS/400) with Domino. This section explores those possibilities.

#### 7.2.1.1 SecureWay directory referrals to Domino

In your SecureWay directory, you can set up default referrals to the Domino Directory. This is configured from the SecureWay Administrator GUI. From the GUI, expand the Server section and then click **Properties**. Put a proper LDAP URL into the Referral field, as shown in Figure 21.



The screenshot shows a window titled "Ready" with a speaker icon. On the left is a sidebar with "General" (selected), "Performance", and "Logging". The main area displays server properties:

<b>Hostname</b>	gefion
<b>Unsecure port</b>	389
<b>Version</b>	<a href="#">IBM SecureWay Directory 3.1</a>
<b>Referral</b>	ldap://

Figure 21. SecureWay default referral setup

Alternately, you could define referrals separately for individual directory entries. This is done by creating an entry that uses the object class referral. Figure 22 on page 112 shows an example of a SecureWay directory that has referrals to an iPlanet directory (for the westerns.com O) and a Domino Directory (for the redbook O).

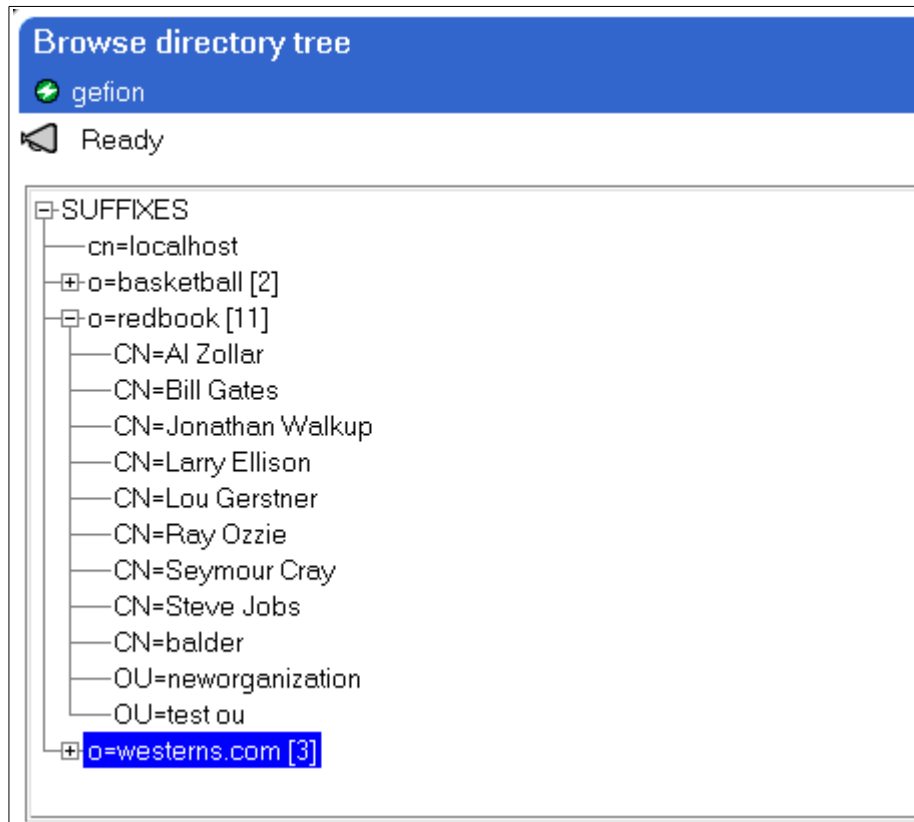
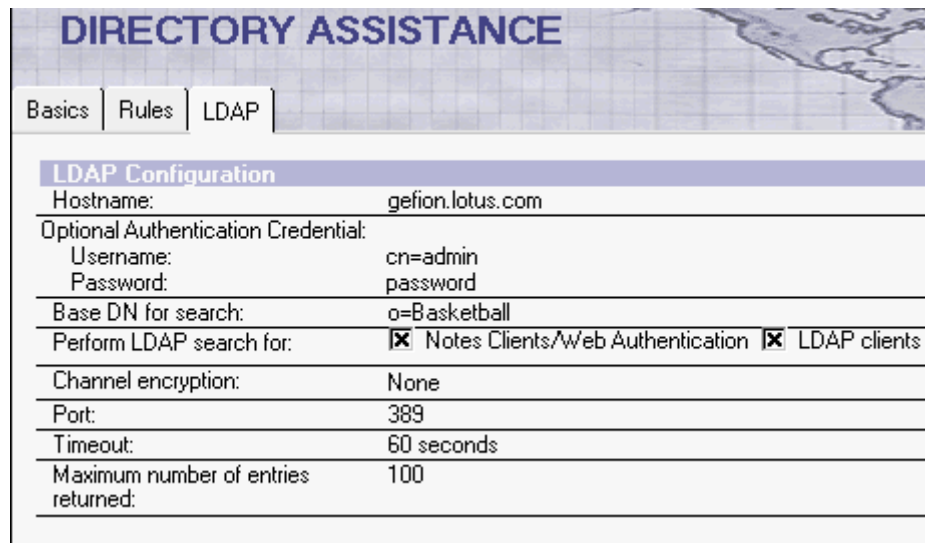


Figure 22. SecureWay directory with referral entries

#### 7.2.1.2 Domino Directory referrals to SecureWay

The SecureWay directory can be configured as an external LDAP directory in Domino using Directory Assistance. Figure 23 on page 113 shows the configuration we used for the SecureWay server. In this case we set up the Directory Assistance matching rule as `*/*/Basketball/*`, trusted for credentials. In addition, we enabled group expansion.



LDAP Configuration	
Hostname:	gefion.lotus.com
Optional Authentication Credential:	
Username:	cn=admin
Password:	password
Base DN for search:	o=Basketball
Perform LDAP search for:	<input checked="" type="checkbox"/> Notes Clients/Web Authentication <input checked="" type="checkbox"/> LDAP clients
Channel encryption:	None
Port:	389
Timeout:	60 seconds
Maximum number of entries returned:	100

Figure 23. SecureWay Directory Assistance configuration

In our sample Domino database, we then added an entry for a group in the SecureWay directory (see Figure 24 on page 114). Since this directory was enabled for group expansion, we were able to log in as a user in that group (Magic Johnson/Players/Basketball) and be authenticated and authorized in that context.

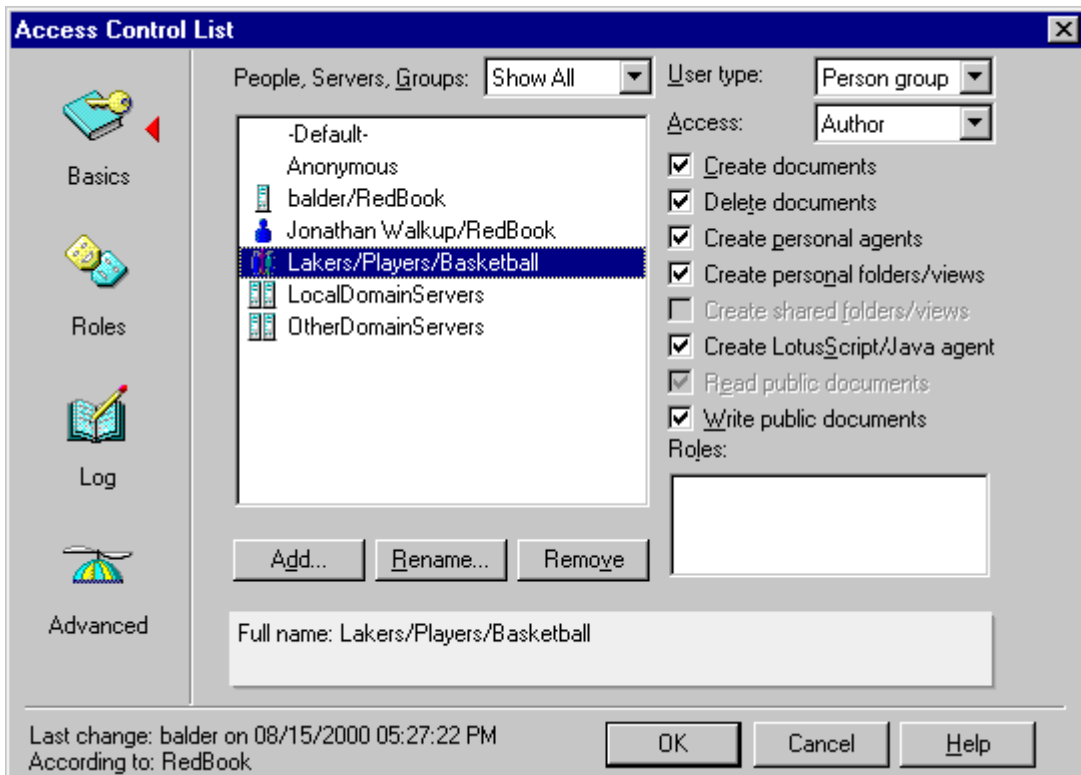


Figure 24. ACL for database with SecureWay group listed

### 7.2.1.3 Websphere with Domino

The WebSphere Application Server is an application engine that runs on top of an HTTP server and provides additional application logic functionality. By integrating it with Domino you can take advantage of the strengths of both platforms. The easiest way to integrate WebSphere and Domino is to have both applications running on the same server. Using this method, WebSphere can use the Domino Directory as its directory.

In order for WebSphere to use the Domino Directory for authentication and authorization, you will need to enable security on the WebSphere server and choose Domino as your LDAP server type. You'll also need to tweak some minor settings to make sure everything works properly. The recently-released redbook *Domino and WebSphere Together*, SG24-5955, covers this installation thoroughly. For detailed instructions, refer to Chapter 6 of that book.

#### 7.2.1.4 LDAPSync

The LDAPSync solution from Lotus Professional Services EMEA addresses the need of bridging third-party LDAP-based directories with Domino Directories.

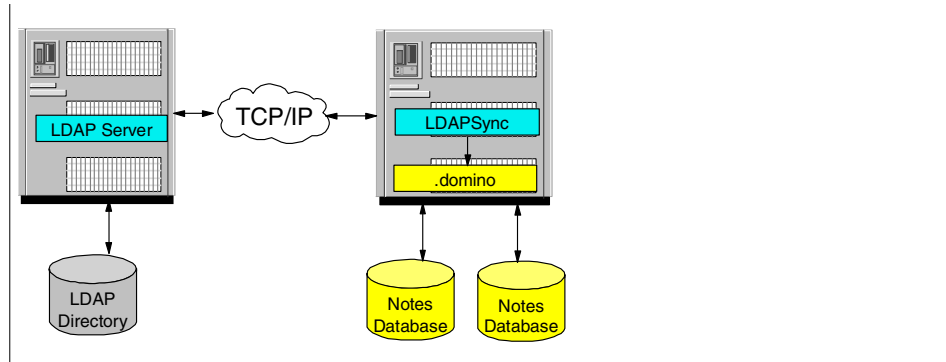


Figure 25. Overview: LDAPSync solution

The LDAPSync solution is a product and service offering based on a set of software that provides synchronization between an LDAP-based directory and a Domino Directory.

The configuration is based on a Notes Database, called the setup database. For each unidirectional synchronization, you create a setup document. This creates a report in a log document, also stored in the setup database.

LDAPSync includes the following utilities:

- LDAPSync is a utility that synchronizes the content of an LDAP directory with the content of a Notes database and vice versa.
- ImportLDIF can be used both to synchronize an LDIF file with Notes database documents (Import) and to export data from a Notes database to an LDIF file.
- SynchroNSF is a utility that synchronizes the content of two Notes databases.
- RunAgent is a utility that runs an Agent (either shared or private) within a Notes database. It is an alternative way of running code located on a Notes database (LotusScript) from a script (.bat file) that is external to the database.

LDAPSync solutions have been tested in various customer situations. For more information, talk to a Lotus Professional Services representative.

## 7.2.2 Microsoft products

Many environments are running Domino on Windows NT, or using Microsoft's Internet Information Server (IIS) or Exchange Server. In these environments, it can be very helpful to consolidate directories as much as possible, to reduce the number of administrative tasks needed, or to reduce the number of passwords that end users need to remember.

### 7.2.2.1 Linking Windows NT user administration with Domino

The Domino user administration process can be linked to the Windows NT/2000 user administration process, so that changes made to one system are made in both. There are three different times that data is synchronized: when a user is registered, when a user is deleted, and when a user's login name or full name changes. While this does duplicate directory entries, it reduces the number of administrative tasks needed to set up users.

To set this up, you need to install the NT directory synchronization services on your Notes client. This will extend the capabilities of the Domino administrator client to also add NT/2000 users and groups, and will extend the NT User Manager to allow the automatic creation of Notes IDs and accounts from that interface.

To install NT directory synchronization services, run the client install program and choose a custom install of the Administrator client. From the list of options, choose **Domino Directory NT Sync Services**, as seen in Figure 26 on page 117.



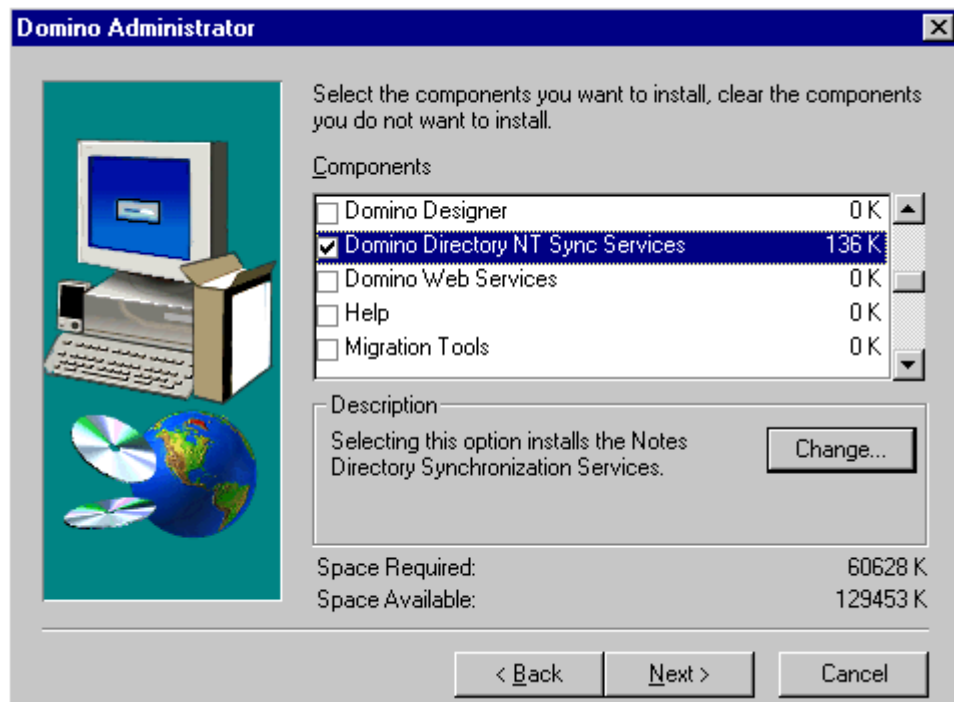


Figure 26. Installing Domino Directory NT Sync Services

Once you've installed the services, you will have a new menu in your Windows NT User Manager. Figure 27 shows an example of this.

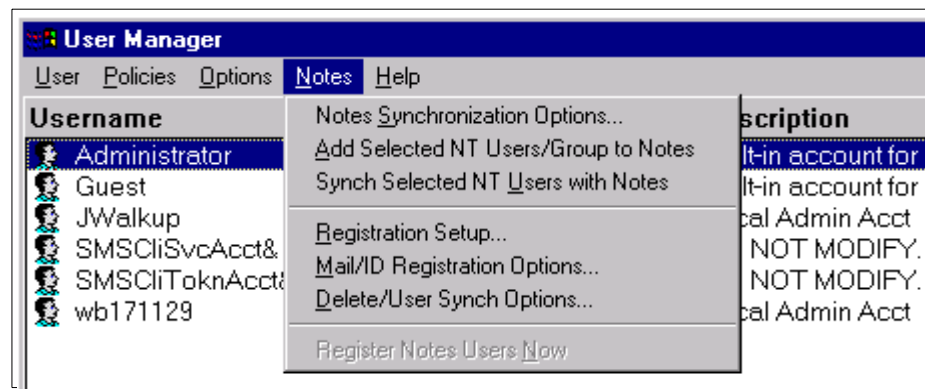


Figure 27. Menu options added to NT User Manager

You should first choose **Registration Setup** to set your defaults for registering Notes users from the NT User Manager. Choosing that option brings up the dialog shown in Figure 28.

Internet registration is useful only if you're not going to be adding these users to a Domino mail system, but you want their entries in the Domino Directory. Use common password will synchronize the user's NT password with their Notes password. Set Internet password in Notes will add that password to the Internet Password field in the Domino Directory. The other options are standard options from the Domino Administrator.

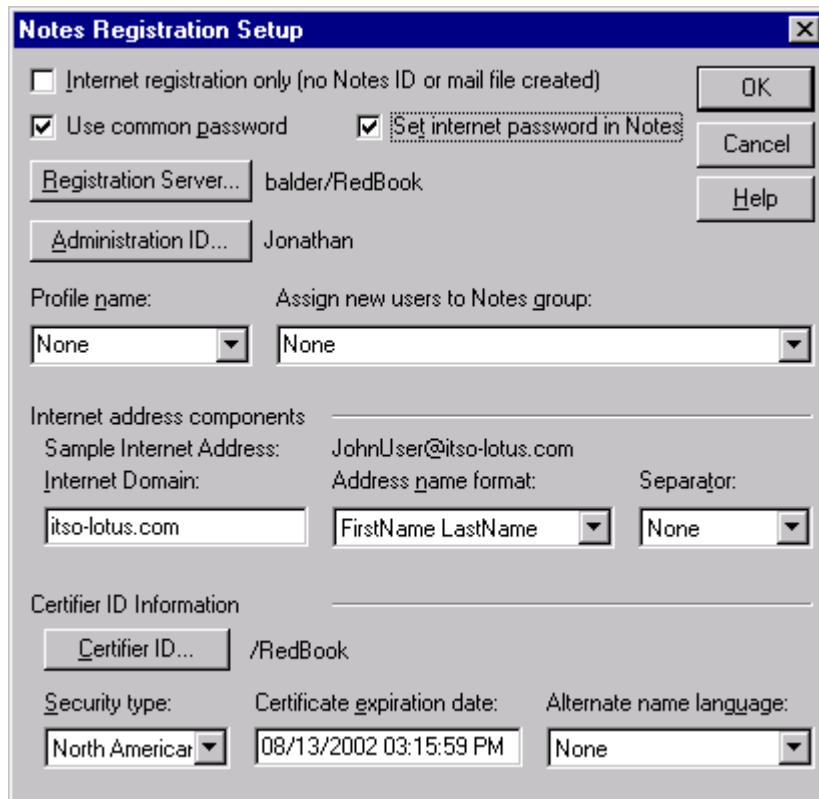
The image shows the 'Notes Registration Setup' dialog box. It has a title bar with a close button. The dialog contains several sections: 1. A checkbox for 'Internet registration only (no Notes ID or mail file created)' which is unchecked. 2. Two checked checkboxes: 'Use common password' and 'Set internet password in Notes'. 3. Two buttons: 'Registration Server...' and 'Administration ID...'. The 'Registration Server...' button is followed by the text 'balder/RedBook'. The 'Administration ID...' button is followed by the text 'Jonathan'. 4. Three buttons on the right: 'OK', 'Cancel', and 'Help'. 5. Two dropdown menus: 'Profile name:' with 'None' selected, and 'Assign new users to Notes group:' with 'None' selected. 6. A section titled 'Internet address components' containing: 'Sample Internet Address:' with 'JohnUser@itso-lotus.com', 'Internet Domain:' with 'itso-lotus.com', 'Address name format:' with 'FirstName LastName' selected, and 'Separator:' with 'None' selected. 7. A section titled 'Certifier ID Information' containing: 'Certifier ID...' with '/RedBook', 'Security type:' with 'North American' selected, 'Certificate expiration date:' with '08/13/2002 03:15:59 PM', and 'Alternate name language:' with 'None' selected.

Figure 28. Notes registration setup

Mail/ID Registration Options allows you to specify where new mail files will be created, where the UserID files will be stored, when the mail files will be created, and some other options about the mail files that are created. Figure 29 on page 119 shows the options you can set in that dialog. If you're going to

be adding people to different servers, you'll need to register them in batches, changing the Mail Server name between batches.

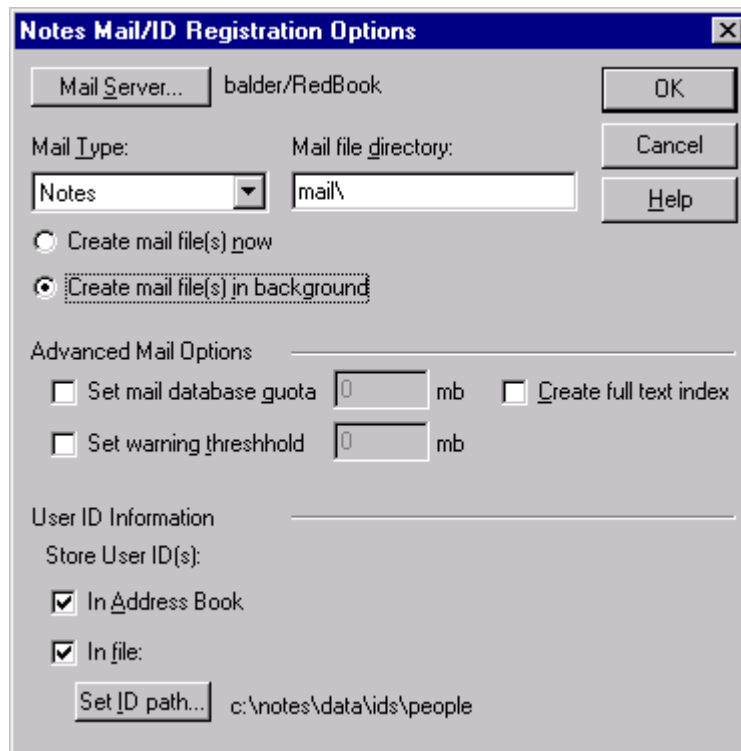


Figure 29. Notes Mail/ID registration options

The final set of options you can configure is under the Delete/User Synchronizations... menu choice. That brings up a dialog like Figure 30. Here you can specify the server to perform the deletions in the address book (typically your administrative hub server), deletion options for the user's mail file, and the server to handle name change synchronizations (again, typically the administrative hub server).

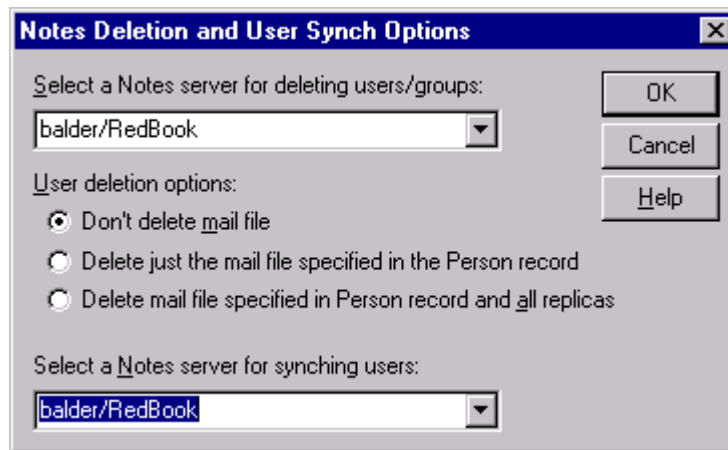


Figure 30. Notes deletion and user synch options

Now, whenever you register new NT users you'll have the option to add them to Domino as well. When you want to add users to the Domino system that have already been set up in Windows NT, choose **Add Selected NT Users/Group to Notes** from the Notes menu. This brings up a mini version of the Register person dialog, as seen in Figure 31.

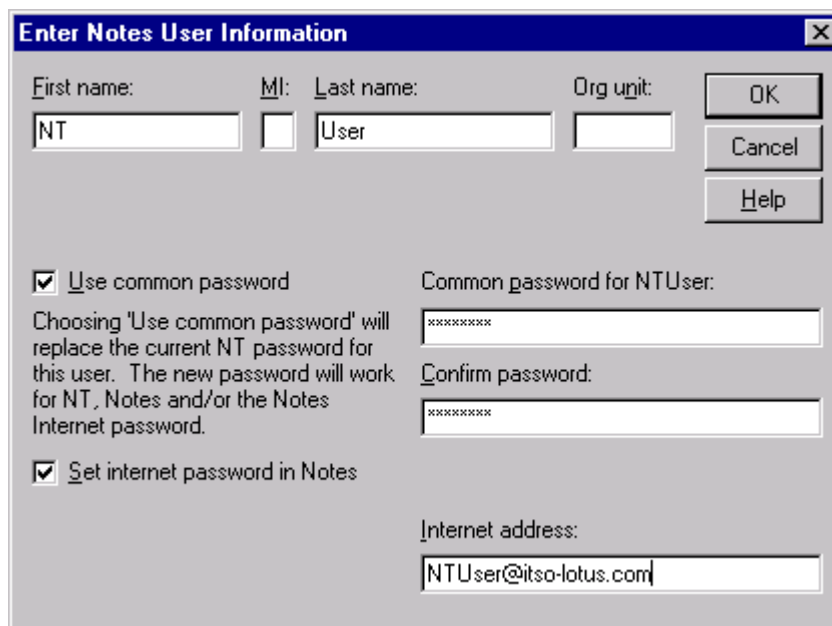


Figure 31. Notes registration dialog

There are also options in the Domino administrator that allow the user being registered to be added to NT at the same time. The relevant section of the User registration dialog is shown in Figure 32.

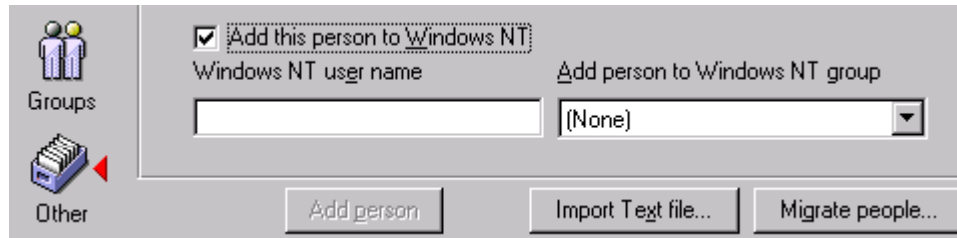


Figure 32. Add user to Windows NT from Domino Administrator

Finally, you can use the Domino Administrator client to migrate users from the Windows NT directory into the Domino Directory. To do this, from the Domino Administrator, choose the **People & Groups** tab, expand the People tools and click **Register... -> Migrate People**. Choose **Windows NT Users/Groups** from the Foreign directory source popup menu and choose the domain you want to use as your source. You should see a dialog like Figure 33 on page 122.

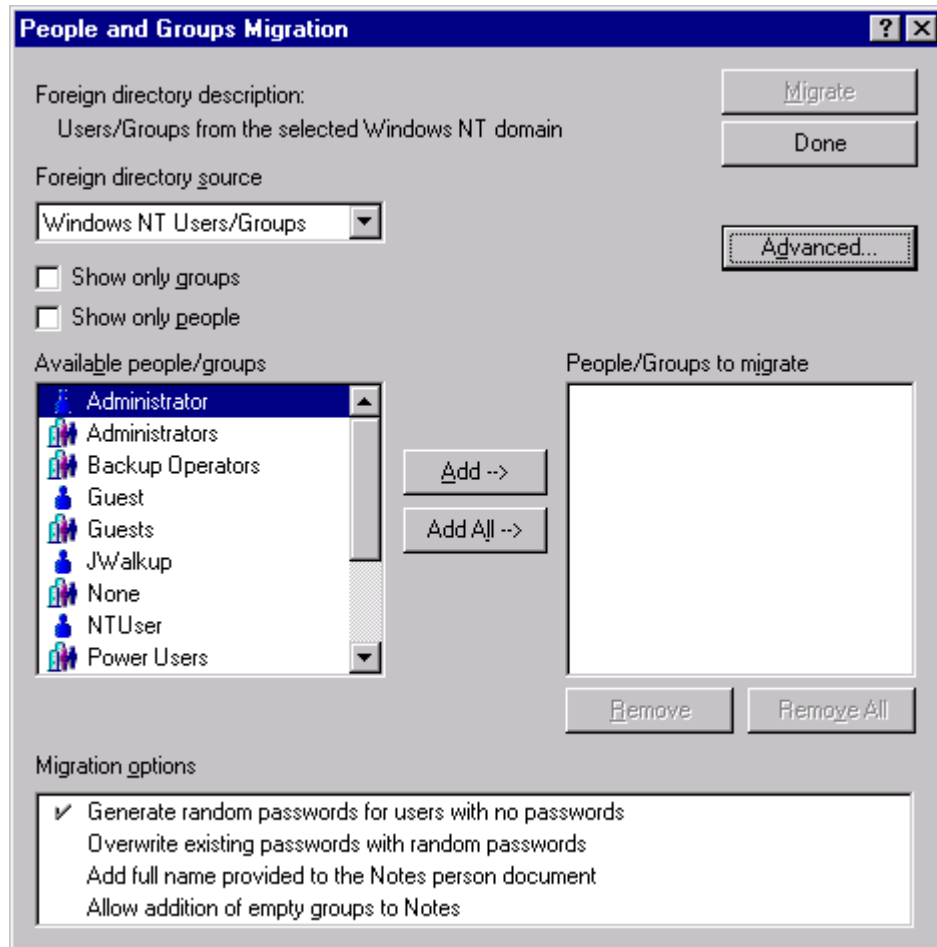


Figure 33. Migrating Windows NT users and groups

Clicking **Advanced...** gives you some additional choices about how to format the User Name field; see Figure 34 on page 123. Here you can tell Notes how to parse the Fullname field in the Windows NT directory into FirstName and LastName fields in Notes. You can also tell the migration process to keep the NT username as the Notes short name.

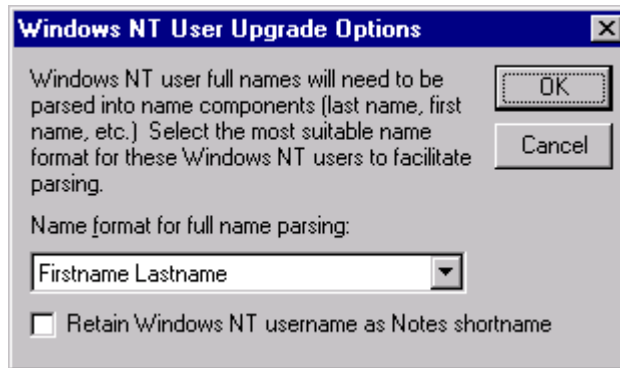


Figure 34. Windows NT user name parsing options

#### 7.2.2.2 Linking Windows NT/2000 client logon with Domino

If your environment consists primarily of Windows NT or Windows 2000 workstations, you can synchronize their NT domain logins with their Notes password, so that they do not have to provide their Notes password. This is accomplished by replacing the gina.dll file that is on the Windows NT workstation with one provided by Lotus. To install this .dll, run a custom install of the Notes client and choose **NT Client Single Logon**, as shown in Figure 35 on page 124. Remember that this only works on Windows NT. Windows 95 and 98 do not have this capability.

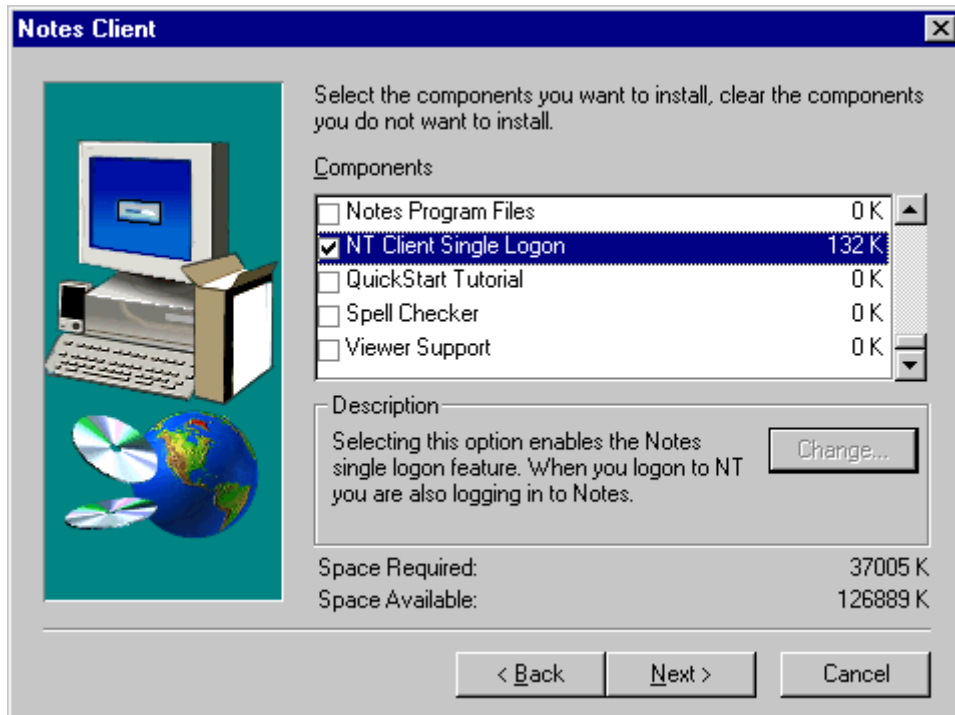


Figure 35. Notes - NT Single Logon install

### 7.2.2.3 Using Active Directory to authenticate Domino users

Since Active Directory is LDAP-compliant, you can set up Domino to use Active Directory to authenticate Web users accessing Domino databases. To do this, you'll need an entry for Active Directory in Directory Assistance. Figure 36 on page 125 shows the Directory Assistance document we used. Because Active Directory's schema is quite different from Domino, we had to use an all-wildcard rule, `*/**/*/*/*`, to match entries.



## DIRECTORY ASSISTANCE

Basics
Rules
LDAP

LDAP Configuration	
Hostname:	heimdal.lotus.com
Optional Authentication Credential:	
Username:	CN=testuser,CN=Users,DC=itso-lotus,DC=com
Password:	password
Base DN for search:	cn=users,dc=itso-lotus,dc=com
Perform LDAP search for:	<input checked="" type="checkbox"/> Notes Clients/Web Authentication <input checked="" type="checkbox"/> LDAP clients
Channel encryption:	None
Port:	389
Timeout:	60 seconds
Maximum number of entries returned:	100

Figure 36. Directory Assistance configuration for Active Directory

You'll also probably want to add entries into ACLs for the Active Directory users. Since Active Directory uses a schema that is very different from what Domino expects, you'll need to specify the attribute names in the ACL entry. See Figure 37 on page 126 for an example. Note the `cn=/cn=/dc=/dc=` format. Since this is different from the Domino schema (`cn/ou/o`), you need to specify the labels. You can still use wildcards, as shown in the example.

Unfortunately, you can't use groups from Active Directory in your Domino ACLs, because Microsoft's LDAP implementation uses a non-standard objectclass for their group entries ("group" as opposed to "GroupOfNames" or "GroupOfUniqueNames"). You could, however, include the Active Directory users in a Domino group to keep the ACL tidy.

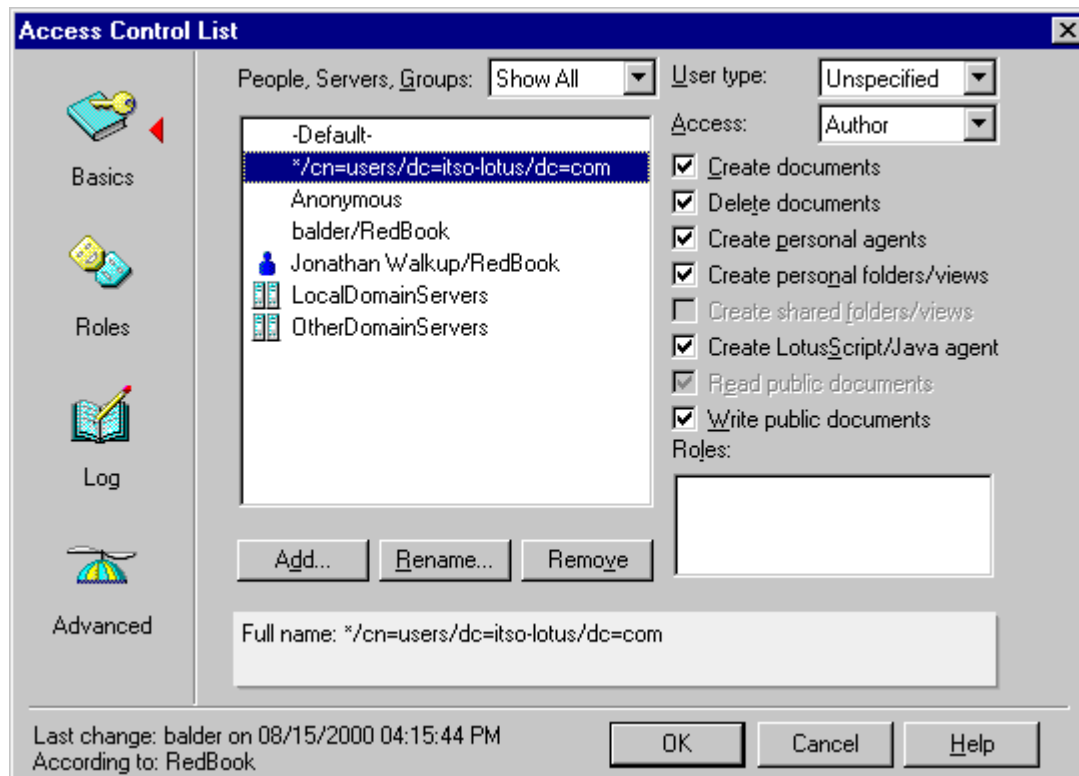


Figure 37. ACL for Active Directory users

#### 7.2.2.4 Linking the Exchange 5.5 directory with Domino

If you have an environment where some users are using Exchange 5.5 for messaging, while others are using Domino, and you want them to be able to address messages between the two environments, you can set up LDAP servers in both environments and use them for addressing information. In Domino, this is configured in Directory Assistance. Figure 38 on page 127 shows an example of our Directory Assistance configuration. The matching rule we used was `*/*/ITSO/*`, trusted for credentials.

LDAP Configuration	
Hostname:	balder.lotus.com
Optional Authentication Credential:	
Username:	
Password:	
Base DN for search:	o=ITSO
Perform LDAP search for:	<input checked="" type="checkbox"/> Notes Clients/Web Authentication <input checked="" type="checkbox"/> LDAP clients
Channel encryption:	None
Port:	3899
Timeout:	60 seconds
Maximum number of entries returned:	100

Figure 38. Directory Assistance configuration for the Exchange 5.5 server

In addition, you can use your Exchange 5.5 directory, exposed using LDAP, to authenticate users to a Domino Web site, via Directory Assistance. Be aware of some caveats, however. First, the Exchange server you are using to authenticate must be a domain controller because of the way that the Exchange server handles authenticated binds. Second, the users' Exchange alias must match their NT account login. We did not have a configuration like this to test, but you should be able to use a Directory Assistance document similar to the one shown in Figure 38.

#### 7.2.2.5 Referring Exchange 5.5 users to a Domino Directory

The Exchange LDAP server can refer searches to an external LDAP directory, which can be a Domino Directory. This is useful if you want to set up an Exchange 5.5 LDAP server that can answer requests about users in the GAL (Global Address List), or refer them to other sources as needed. You can set LDAP referral for a single server or an entire exchange site. See Figure 39 on page 128 for an example of the configuration for an entire site.

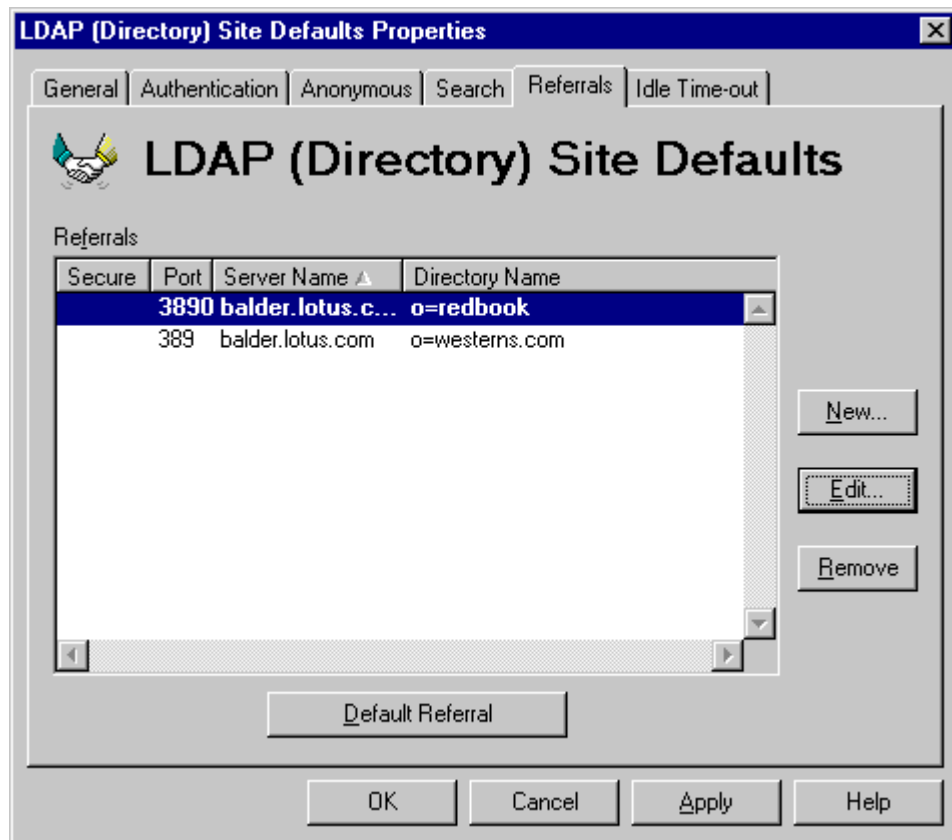


Figure 39. Exchange 5.5 LDAP referral setup

You can also set up outlook clients to do direct searches against an LDAP directory. This is set up by adding an additional service to the Outlook configuration, Microsoft LDAP Service. Figure 40 on page 129 shows a sample configuration to point to a Domino LDAP directory.



Figure 40. Outlook LDAP directory service setup

#### 7.2.2.6 Migrating users from Exchange 5.5 directories

If you are faced with the prospect of moving users from Exchange 5.5 to Domino, you can use the Domino Administrator client to ease the transition. In the Domino Administrator, switch to the **People & Groups** tab, expand the People section under Tools, and click **Register...** Enter your certifier password and you will be presented with the Register Person dialog box. Click **Migrate people...** and choose **Exchange Users** from the Foreign directory source menu. You will have to have the Exchange Administrator client installed on your machine to perform the migration. Once you've selected Exchange Users, you will see a dialog like Figure 41 on page 130.

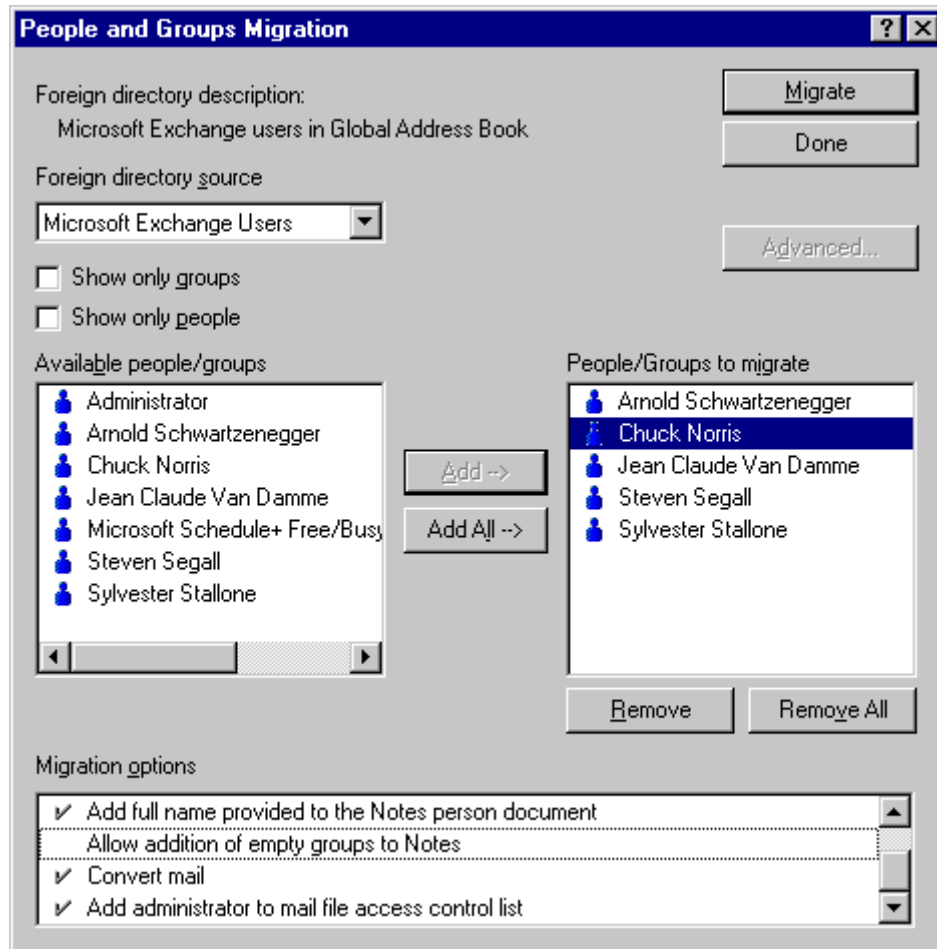


Figure 41. Exchange user migration dialog

Choose the users you want to move and click **Migrate**. They will be added to your registration queue. You've got a couple of options that you didn't have in the NT migration dialog. "Convert mail" will move the user's exchange mailbox into their Domino mail file. "Add administrator to mail file access control list" will give you access to the individuals' mail files.

#### 7.2.2.7 Linking Exchange 2000 with Domino

Since Exchange 2000 uses Active Directory to store its directory information, and Active Directory uses LDAP technology at its heart, you can use Directory Assistance on a Domino server to provide mail addressing lookup.

The Directory Assistance document in that case would look like Figure 36 on page 125.

There is currently no method for migrating users from an Exchange 2000 installation to Domino, but this function should be included in the next feature release of Domino.

#### 7.2.2.8 Linking IIS with Domino

One of the new features of Domino 5.0 was the ability to run IIS as the HTTP engine for Domino. This works with IIS version 4.0. To do this, you have to have both servers installed, then modify the IIS configuration so that it hands off any .nsf URLs to the Domino Web server engine. Additionally, the synchronization links username/password authentication between the two systems.

To set this up, you set up a Domino server as you normally would, but you do not run the HTTP task. This will conflict with IIS. Instead, you need to configure IIS to pass off requests for .nsf files to the Domino engine. To do this, you need to configure IIS using the Internet Service Manager. From the Internet Service Manager, right-click on your Web site and select **Properties**. Select the **Home Directory** tab and click **Configuration -> Add**. Fill out the dialog as in Figure 42, replacing the pathname to the niisextn.dll if necessary. Click **OK** to add your change to the extension mapping list.

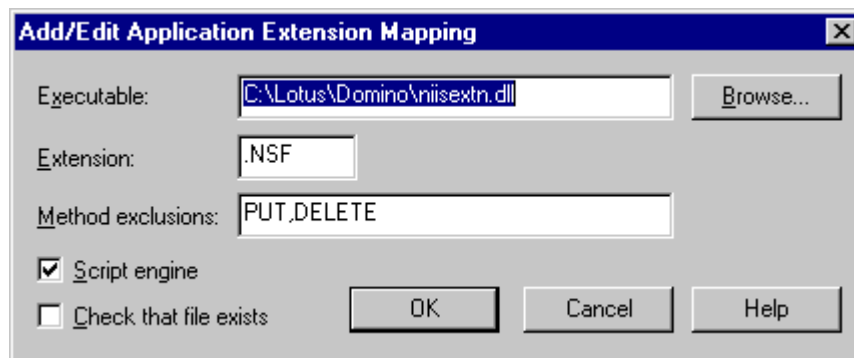


Figure 42. Adding Domino extension mapping to IIS

Now you need to add the Domino ISAPI filter. To do this, click the **ISAPI Filters** tab. Click **Add**. Now add some descriptive text about the filter, and the pathname to the file. Use Figure 43 on page 132 as a reference. Now click **OK** to add the Domino ISAPI filter to the IIS configuration.

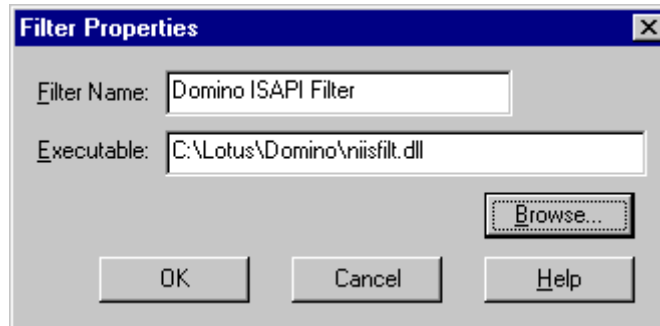


Figure 43. Installing the Domino ISAPI filter in IIS

Now you can close the Web server properties box and move to the final step in configuring IIS for use with Domino: Setting up virtual directories for icons and Java applets. To do this, right-click on the Web server icon and choose **New -> Virtual Directory**. This will start the virtual directory wizard. You will run this wizard twice, once for the icons directory and once for the Java directory. For the icons directory, use an alias of *icons*, and the path to the icons subdirectory (<notesdata>\domino\icons). For the Java directory, use *domjava* as the alias and the path to the applet directory (<notesdata>\domino\java). You can accept the default permissions for each directory.

Once you've set up IIS with Domino, you may need to adjust your Domino Directory so that the credentials passed by IIS to Domino match entries in your directory. If you are using Basic Authentication (username/password), Domino will verify the username passed to it by IIS, and match it to a person document in the Domino Directory, but won't validate the password against the HTTP password field. If you are using NTLM authentication (Windows NT Challenge/Response), Domino will use the same procedure, but you will need to add the NT user's domain credentials to their username field. The username should be specified in the format <NT domain>\<userid>. Again, Domino will not check the Internet password field to verify the password that the user submitted.

#### 7.2.2.9 Using Outlook clients with Domino servers

If you are using Outlook clients on the desktop, you can install a MAPI provider for Notes that will give you access to a mail file on a Domino server. You will still need to have the Notes client installed on the workstation, but you can use Outlook to read your mail. To install this piece, you will need to install Notes after Outlook has been installed. Then in Outlook, choose **Tools -> Services** and click **Add**. One of your options should be "Lotus Notes



Mail". Select it and enter your password. Now you should be able to access your Domino server using Outlook.

A new Lotus product, iNotes access for Outlook, was announced in January at Lotusphere. This product provides connectivity for Outlook clients without having to install the Notes client and MAPI service provider. It also provides the ability to use Outlook offline. For more information, see

<http://www2.lotus.com/home.nsf/welcome/inotes>.

#### **7.2.2.10 Importing LDIF data into Exchange 5.5**

Exchange 5.5 doesn't natively support LDIF importing. One possible solution is to import your LDIF file into an Outlook Express address book and then export it to .csv format (which Exchange 5.5 can import). We were unable to successfully import an LDIF file into Outlook Express, but it is theoretically possible.

#### **7.2.2.11 Importing LDIF data into Active Directory**

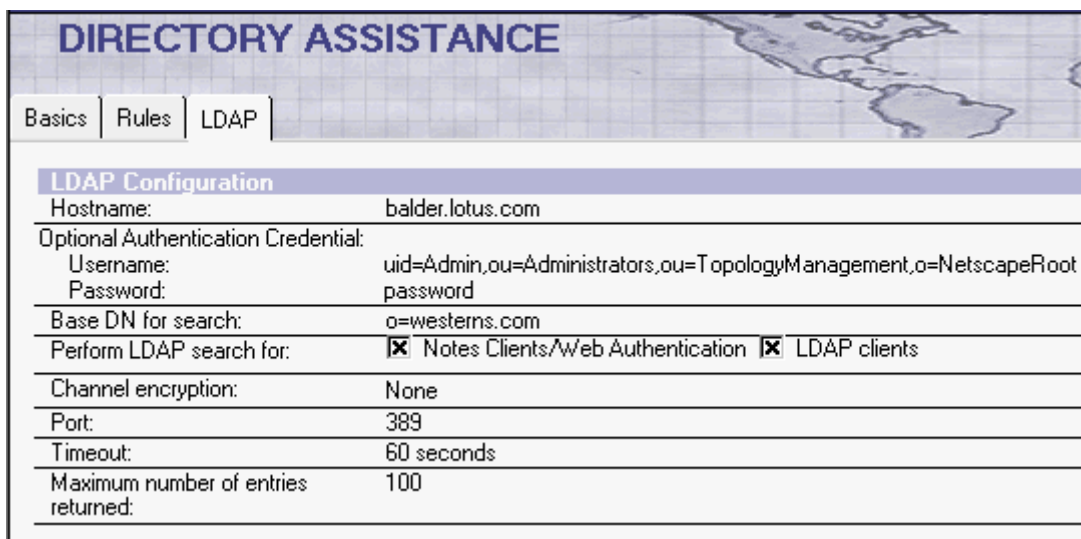
With the release of Active Directory, Microsoft has embraced the LDIF standard. They bundle a utility with Windows 2000, `ldifde.exe`, that imports directory data in LDIF format into Active Directory.

## 7.2.3 iPlanet (Netscape/Sun) products

The Sun-Netscape alliance was created shortly after America OnLine bought Netscape. The two companies jointly promote a line of products under the iPlanet brand. The two main products we're concerned with are the iPlanet directory server and the iPlanet Web server, both based on the former Netscape products.

### 7.2.3.1 Domino Web server with iPlanet directory server

In order to use an iPlanet directory server for authentication to a Domino web site, you need to set up a Directory Assistance document pointing to the iPlanet web site. Then you need to specify the DNs that iPlanet returns in the ACL of the databases that you wish to secure. Figure 44 shows an example of the LDAP tab on the Directory Assistance configuration document that we set up. The matching rule we used was `*/*/*/westerns.com/*`, trusted for credentials.



The screenshot shows the 'DIRECTORY ASSISTANCE' configuration window with the 'LDAP' tab selected. The 'LDAP Configuration' section contains the following settings:

LDAP Configuration	
Hostname:	balder.lotus.com
Optional Authentication Credential:	
Username:	uid=Admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot
Password:	password
Base DN for search:	o=westerns.com
Perform LDAP search for:	<input checked="" type="checkbox"/> Notes Clients/Web Authentication <input checked="" type="checkbox"/> LDAP clients
Channel encryption:	None
Port:	389
Timeout:	60 seconds
Maximum number of entries returned:	100

Figure 44. Directory Assistance configuration for iPlanet LDAP server

Figure 45 on page 135 shows a sample Domino ACL with an entry from the iPlanet server. Notice that the format has been converted from the LDAP format (`cn=,ou=,o=`) to the Notes format (`cn/ou/o`). Also note that we chose to use wildcard characters instead of listing individuals by name.

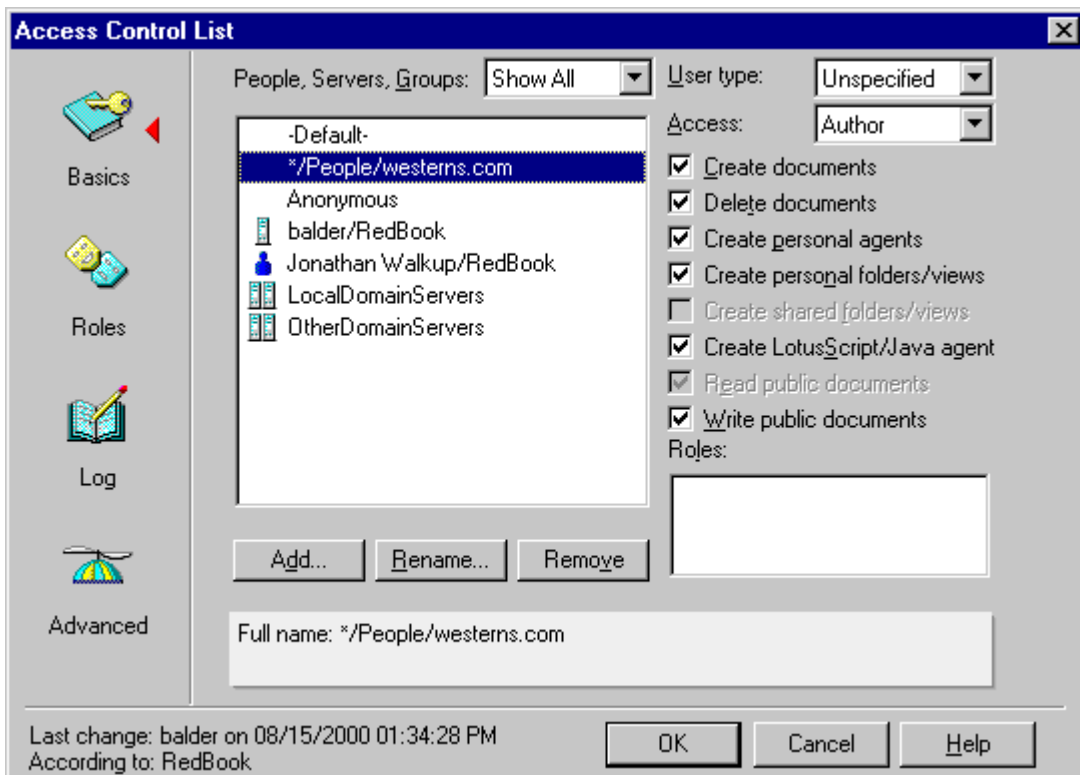


Figure 45. Domino ACL with an entry from an iPlanet server

### 7.2.3.2 Domino Directory referrals to iPlanet directory server

You can also use a Directory Assistance document like the one shown in Figure 44 on page 134 for mail addressing to Notes clients and referrals to LDAP clients. Remember that external LDAP directories will not appear in the Addressing dialogue unless separately configured using Account documents in the Personal Address Book, and will not be consulted during "type ahead." An external LDAP client will receive a referral to the original source directory.

### 7.2.3.3 iPlanet directory server referrals to Domino server

When setting up referrals from an iPlanet directory server, you can choose to set up one server for default referrals, referral servers for individual entries in the Directory, or both.

To set up default referrals from an iPlanet directory server to a Domino server, use the Netscape Admin Console. From the console, choose the **Configuration** tab, then the **Services** tab. You should see a screen as shown

in Figure 46. Enter an LDAP URL that points to your Domino server and click **Save**. If you have multiple servers that you want to use for referrals, you can enclose each URL in quotes and separate them with spaces. See RFC 2255 in B.2.1, “Core Specifications” on page 160 for the syntax of an LDAP URL.

The screenshot shows the 'Settings' tab of the iPlanet Directory Server configuration. Under 'Network Settings', the 'Port' is set to 389, 'Encrypted port' is 636, and 'Referrals to' is ldap://balder.lotus.com:3890. The 'NT Synchronization Service' section has 'Enable NT Synchronization Service' unchecked, 'Use SSL in NT Synchronization Service' checked, and 'Synchronization port' set to 5009. At the bottom, 'Make entire server read-only' is unchecked, 'Track entry modification times' is checked, and 'Enable schema checking' is checked. 'Save', 'Reset', and 'Help' buttons are at the bottom right.

Figure 46. iPlanet Directory Server Referrals configuration

If you want to set up a referral for a specific entry in the directory, what is referred to as a “Smart Referral” by the iPlanet server, you do that by adding the object class reference to the entry, and adding a valid LDAP URL to the ref attribute. You would do this if you needed to refer to a particular server for the authoritative information about an individual. Again, you can use the Administration Console to set this up.

First, choose the **Directory** tab and the entry that you wish to modify. Double-click that entry and click **Advanced....** From the View menu, choose **Show all attributes**. Scroll down and right-click on the **ObjectClass** attribute in the list. Choose **Add Value** from the popup menu that appears. Scroll down

that list and choose **referral** from the list of available object classes to add. Click **OK** to add it to the entry. Now scroll down to where the attribute ref has been added and right-click on it. Choose **Add Value** from the popup and enter the LDAP URL for your referral. Your final result should look like figure Figure 47. Click **OK**, then **OK** on the edit entry screen to save your changes.

The screenshot shows a 'Property Editor' window for a user named 'John Wayne'. The window has a menu bar with 'File', 'Edit', and 'View'. The main area contains a list of attributes and their values. The 'Object class' attribute is currently selected, and its list is expanded to show the following options: 'top', 'person', 'organizationalPerson', 'inetOrgPerson', and 'referral'. The 'ref' attribute is set to the LDAP URL 'ldap://baldern.lotus.com:3890'. Other attributes visible include 'Mobile phone number', 'Organization', 'Organizational Unit', 'P.O. box', 'Pager number', 'Password' (masked with asterisks), 'Photograph', and 'Registered address'. At the bottom right, there are 'OK', 'Cancel', and 'Help' buttons.

Figure 47. Adding a Smart Referral to an iPlanet directory entry

#### 7.2.3.4 iPlanet Web server with Domino Directory server

To set up the iPlanet Web server to use a Domino Directory server for user authentication, you have two options. First, you can set the Domino server as the default server. Alternately, you can define the Domino server as an

alternate directory server and use it in ACLs that you apply to the Netscape server.

To set your Domino server as the iPlanet Web server's default directory server, use the iPlanet Administration server. In the Administration server, click the **Global Settings** tab and then the **Configure Directory Service** link. Fill in the screen provided with the hostname and port of your Domino LDAP server, a base DN for searching, and a bind DN and password to access the directory. See Figure 48 for an example of how the configuration should look. Now in your iPlanet ACLs you can use users and groups from the Domino Directory.

**Configure Directory Service**

**LDAP Directory Server Configuration**

Host Name:

Port:

Use Secure Sockets Layer (SSL) ☐ Yes ☒ No

for connections?:

Base DN:

Bind DN :

Bind Password :

Figure 48. External LDAP directory configuration for iPlanet Web server

If you like, you can define the Domino server as an alternate directory to be used for authentication. To do this, you should manually edit the `dbswitch.conf` file in the `<serverroot>/userdb` directory. Here's an example of how this file might look.

```
directory default ldap://balder.lotus.com:3890/o%3Dredbook
default:binddn cn=jonathan walkup,o=redbook
```

```
default:encoded bindpw cGFzc3dvcmQ=  
directory heimdal ldap://heimdal.lotus.com:3890/o%3DITSOCERT  
heimdal:binddn cn=Something Easy,o=ITSOCERT  
heimdal:pindpw password
```

Here is an example of how you would include an entry from that directory in an iPlanet ACL:

The screenshot shows a configuration window titled "User/Group". It contains several radio buttons and text input fields. The "Authenticated people only" option is selected. Under this, "Only the following people" is selected, with "Group" and "User" (containing "ejohn") fields and "List" buttons. There are also fields for "Prompt for authentication", "Authentication Methods" (with "Default" selected), and "Authentication Database" (with "heimdal" selected in a dropdown). At the bottom are "Update", "Reset", and "Help" buttons.

Figure 49. iPlanet Web server ACL using an external directory

#### 7.2.3.5 Importing LDIF data into iPlanet directory server

You can import LDIF data generated by Domino into an iPlanet directory server. This operation is performed using the Netscape Console. While working with the directory server that you want to import the data into, switch to the **Configuration** tab and choose **Console -> Import**. You should see a dialog as shown in Figure 50 on page 140. Type or browse to the name of the file that you want to import, select any options you want and click **OK**. The LDIF data will be added into the iPlanet directory server.

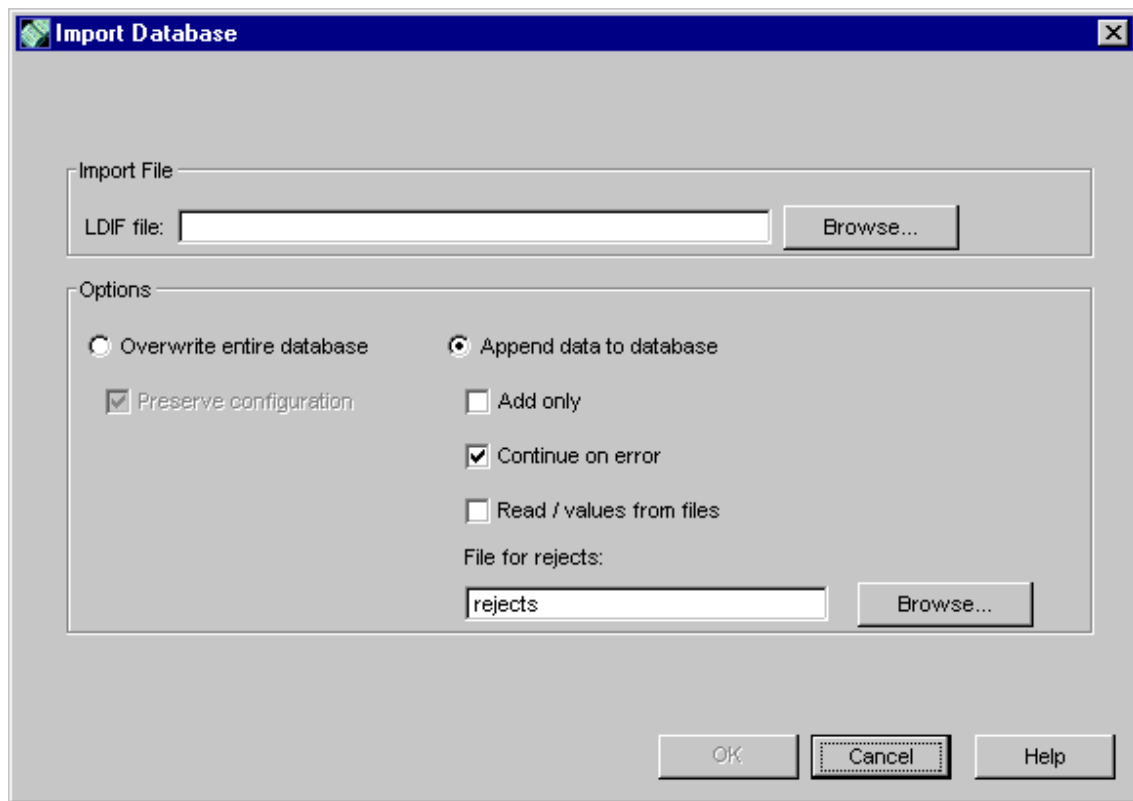


Figure 50. Importing an LDIF file into iPlanet directory server

## 7.2.4 Novell products

Here are a few examples of Novell products with which Domino interacts.

### 7.2.4.1 Domino server referrals to eDirectory server

As in previous examples, to set up your Domino Directory to use an eDirectory server for mail lookups and Web authentication, you need to set up a Directory Assistance configuration document. An example of the Directory Assistance document that we used is shown in Figure 51 on page 141. The matching rule we used was `*/*/redbooknds/*`, trusted for credentials.



## DIRECTORY ASSISTANCE

Basics
Rules
LDAP

LDAP Configuration	
Hostname:	odin.lotus.com
Optional Authentication Credential:	
Username:	cn=Admin,o=RedBookNDS
Password:	password
Base DN for search:	o=RedBookNDS
Perform LDAP search for:	<input checked="" type="checkbox"/> Notes Clients/Web Authentication <input checked="" type="checkbox"/> LDAP clients
Channel encryption:	None
Port:	389
Timeout:	60 seconds
Maximum number of entries returned:	100

Figure 51. Directory Assistance document for eDirectory server

In this example, we have set up a group in the Domino Directory that contains names from the LDAP source. We'll use that group in the ACL of the test Domino database. Figure 52 shows the group and Figure 53 on page 142 shows the ACL settings we used.

## GROUP: eDirectory Users

Basics
Administration

Basics:	
Group name:	eDirectory Users
Group type:	Multi-purpose
Description:	Group of names from eDirectory
Members:	bcosby/RedBookNDS eddie/RedBookNDS rwilliams/RedBookNDS smartin/RedBookNDS swright/RedBookNDS

Figure 52. Group of eDirectory users

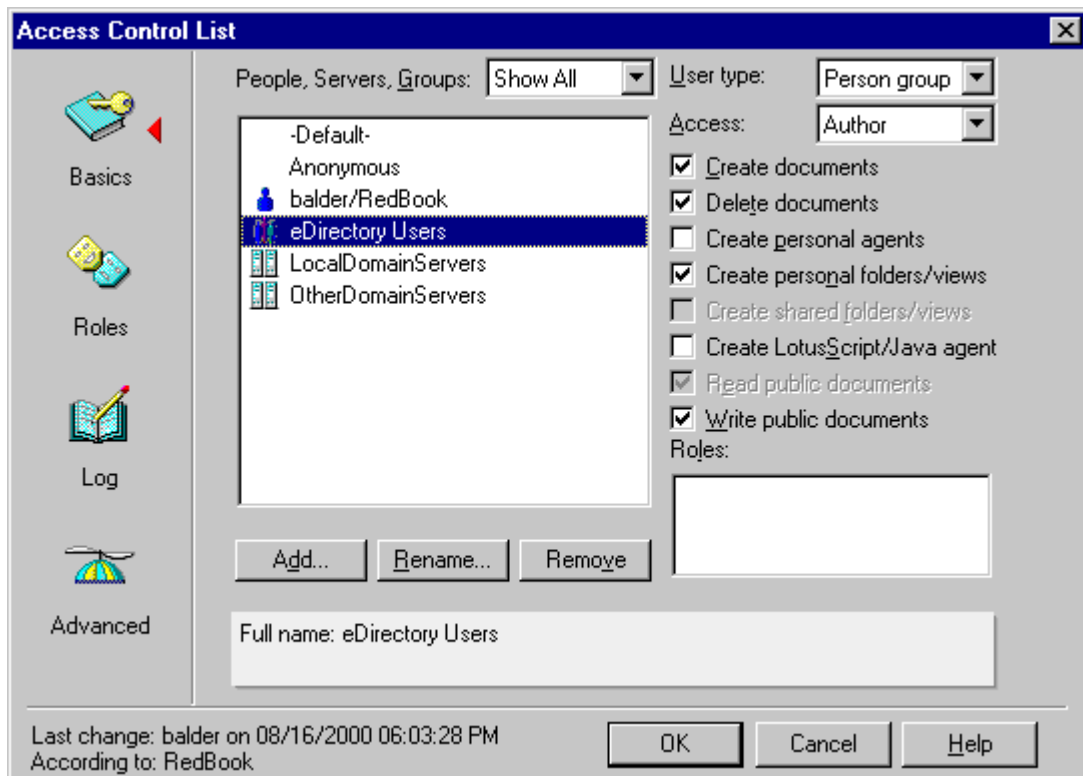


Figure 53. ACL of database for eDirectory users

#### 7.2.4.2 eDirectory referrals to Domino server

To set up your eDirectory server so that it will hand off referrals to a Domino Directory server, you need to use the ConsoleOne application. You can set up referrals for a group of LDAP servers so that any of them will refer requests not in their directory to another directory. Figure 54 on page 143 shows an example of how we set up referrals on an eDirectory server.

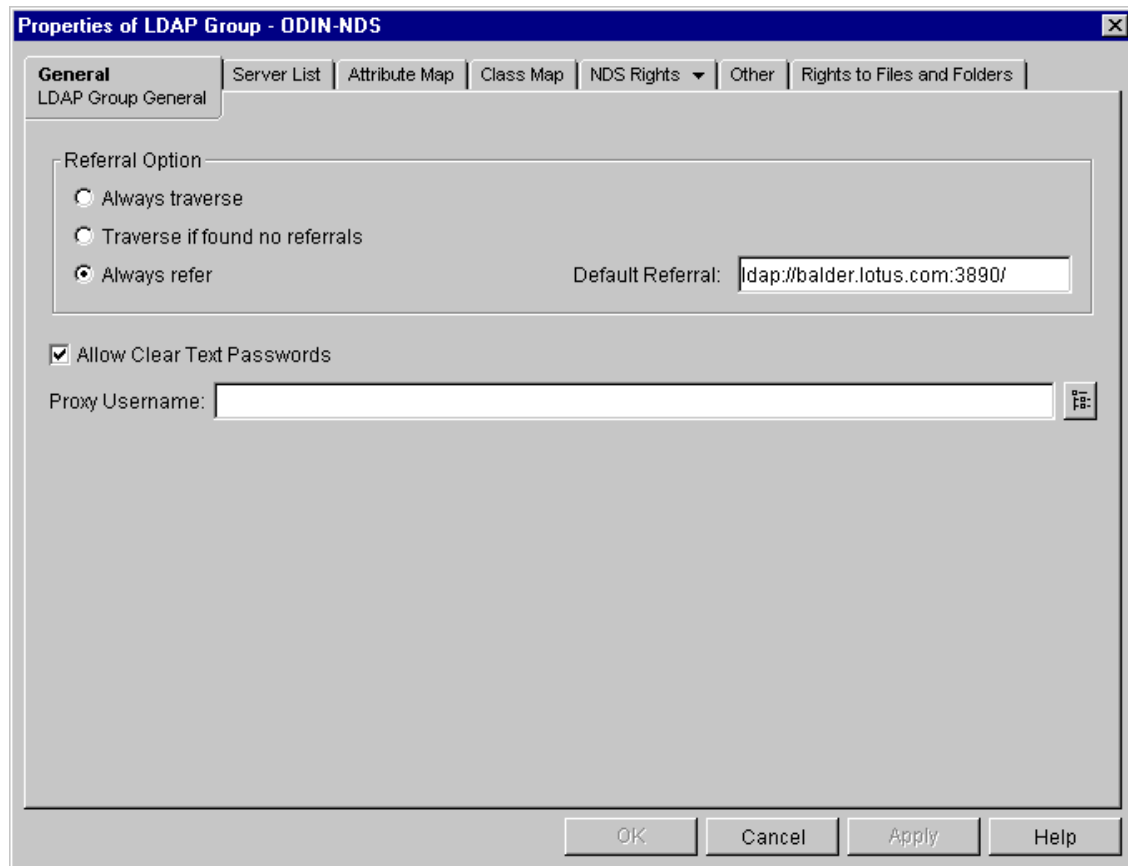


Figure 54. Referral setup in eDirectory

## 7.2.5 Other LDAP directories

There are a number of other LDAP directory products available or in use today. In general, Domino can refer to these directories using Directory Assistance for mail lookup or authentication to Domino Web sites. If you're running an LDAP server that is not LDAPv3-compliant, be aware that you may need to change some of the handshake parameters in your Directory Assistance document to make communication between the servers work smoothly.

### 7.2.5.1 Innosoft

Innosoft Inc.'s Innosoft Distributed Directory Server (IDDS), along with the company itself, was purchased by Sun/iPlanet in March of 2000. They are no

longer selling IDDS, but are planning on incorporating its functionality into version 5.0 of the iPlanet Directory server. For more information, see

<http://www.innosoft.com/>

#### **7.2.5.2 OpenLDAP**

OpenLDAP is a freeware, open source LDAP server. It is built upon earlier work by the University of Michigan. Version 2.0, which supports LDAPv3, is currently under development. Status updates and downloads can be found at

<http://www.openldap.org/>

### **7.2.6 Metadirectory products**

Metadirectory products are designed to integrate various directory systems. In this section we show how two of the most popular metadirectory products are configured to use the Domino Directory as a data source.

#### **7.2.6.1 DirX**

Siemens' DirX server is an x.500 product, but there is a metadirectory component, DirX Meta Directory, that can connect directly to a Domino Directory and synchronize that directory with the DirX x.500 directory. Connectors are available for other directory products as well.

DirX is a complex product, and much of the configuration is handled with TCL scripts and command-line utilities. The meta hub pulls data from external sources and creates files that can be imported into the x.500 directory, and vice-versa. A script can be written to pull entries from (or push entries into) a Domino Directory. For more information about DirX and DirX Meta Directory, see

<http://www.ic.siemens.com/networks/gg/isa/md/ps.htm>

#### **7.2.6.2 MMS**

Microsoft Metadirectory Services (MMS) is a successor to the Zoomit metadirectory product. It is set up on a Windows 2000 server, and can also connect directly to a Domino Directory server.

To get MMS and Domino linked, we installed the MMS server code as well as MMS Compass version 2.2. In both cases, we used the default configuration settings. In addition, we also installed the Lotus Notes Add-in, which is available with the MMS code. For the MMS connection to the Domino server to succeed, you have two options:

- Run both the Domino and MMS server on the same server.

- Provide the server running MMS with file system access to the server.id in the notes\data directory on the Domino server, by mapping a network drive from the MMS server. Ensure that the drive is included in the path statement of the server's environment settings.

We chose the first option as our approach. In our test environment, the NT domain name is "itso-lotus" and the metadirectory domain name is "itso-meta" with the local servername of "Heimdall" and a Notes domain name of "Redbooks".

Figure 55 shows our directory registry before we set up the Domino synchronization.

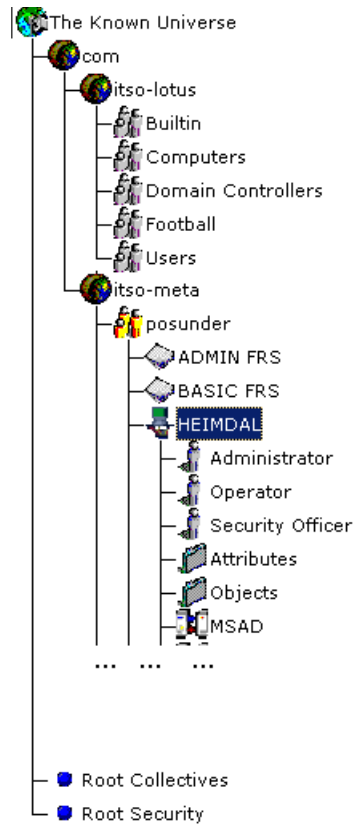


Figure 55. Directory registry before synchronization

The next step was to create a management agent. These agents are used to automate update processes between MMS and the target directories. The

option to create this agent is available only when you select the MMS server as shown in Figure 56.

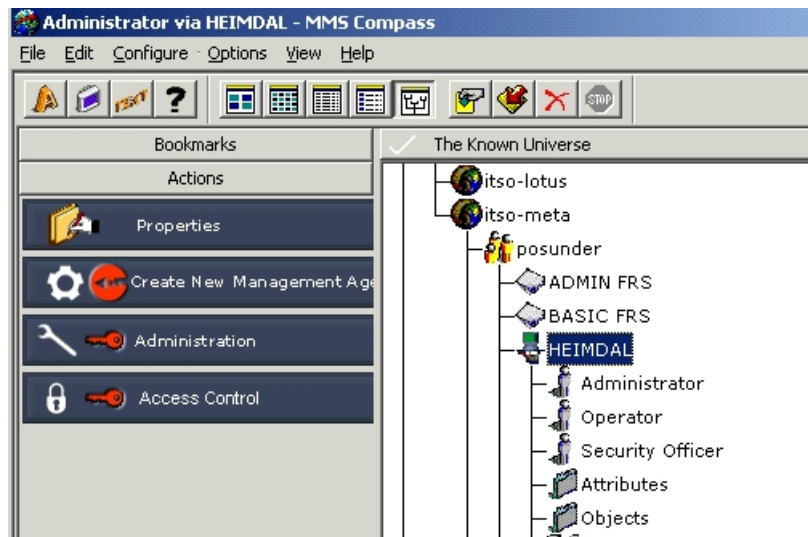


Figure 56. Selecting the MMS server

When you click on the button to create a new management agent, you will get the interface shown in Figure 57.

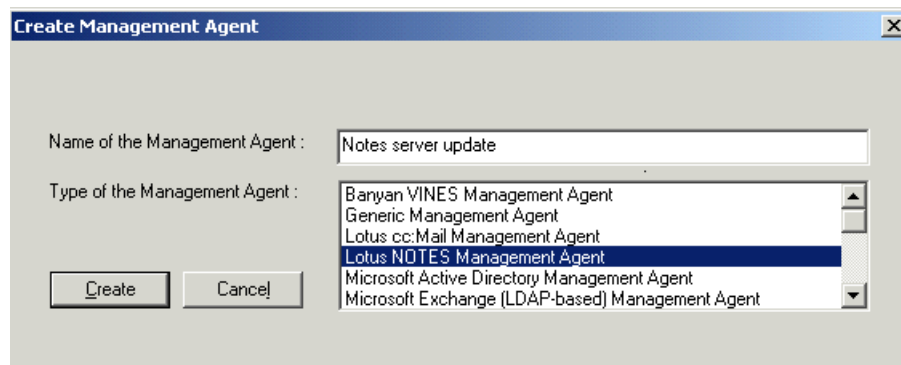


Figure 57. Creating a management agent

Once the server has created the agent, you will see the screen in Figure 58.



Figure 58. Beginning to configure the management agent

The default domain where the agent will add the information is the meta domain, in this case, "itso-meta.com". The following tab is where you need to specify the Notes information. The default address book used is "names.nsf", so you only need to specify a Domino Directory if you are not going to use it. Once you have added the Domino server name in, click **Test your connection** to verify that the management agent can connect to the Domino server. See Figure 59 on page 148.

**Configure the Management Agent**

Connected Directory Specifics | Metadirectory Relationships | Personal Names | Inclusions and Exclusions

Mode and Namespace Management | **Discovery** | Advanced Discovery Parameters | Foreign Users | Ne

Discovery Parameters

Notes registration server name:   
e.g. notes1/org1

Administrator password:

Main Address Book

Name of the domain associated with the Main Address Book:   
(Optional - the name of the address book will be used by default)

Figure 59. Completing the configuration of a management agent

The other default setting within the Lotus Notes management agent is for the MMS server to only pull the information out of the Domino Directory. To enable a two-way update process, you need to set the attribute flow settings for the management agent. Again, this button is only available when you have selected the agent on which these rules will apply, which enables you to update multiple Domino directories with different configurations. See Figure 60 on page 149.



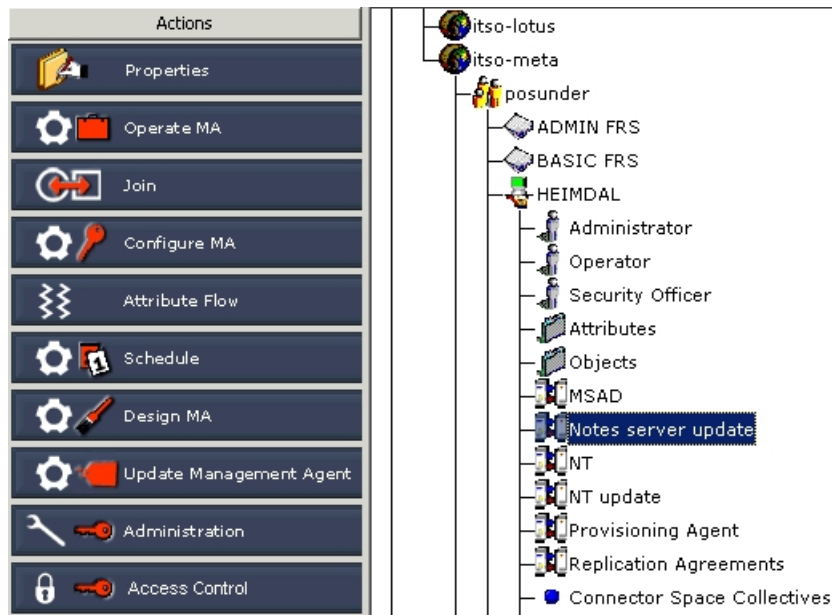


Figure 60. Updating server information

In our tests, we used the Domino Directory mobile and pager numbers as the master information. In addition, we used the company and manager name from the MMS metadirectory as master information. See Figure 61 on page 150.

**Note:** Not all the fields between the Domino Directory and the MMS metadirectory map by the names shown. You will have to do some testing to ensure the correct mapping for the fields that you need.

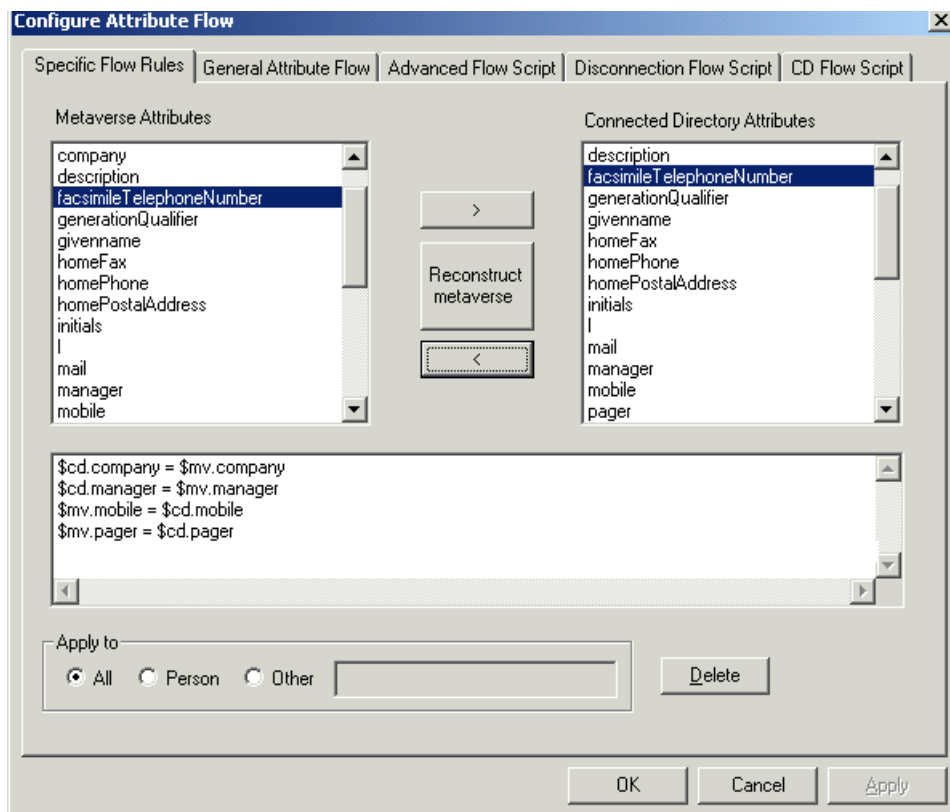


Figure 61. Master information

Figure 62 is an example of the log created when you run the Lotus Notes management agent.

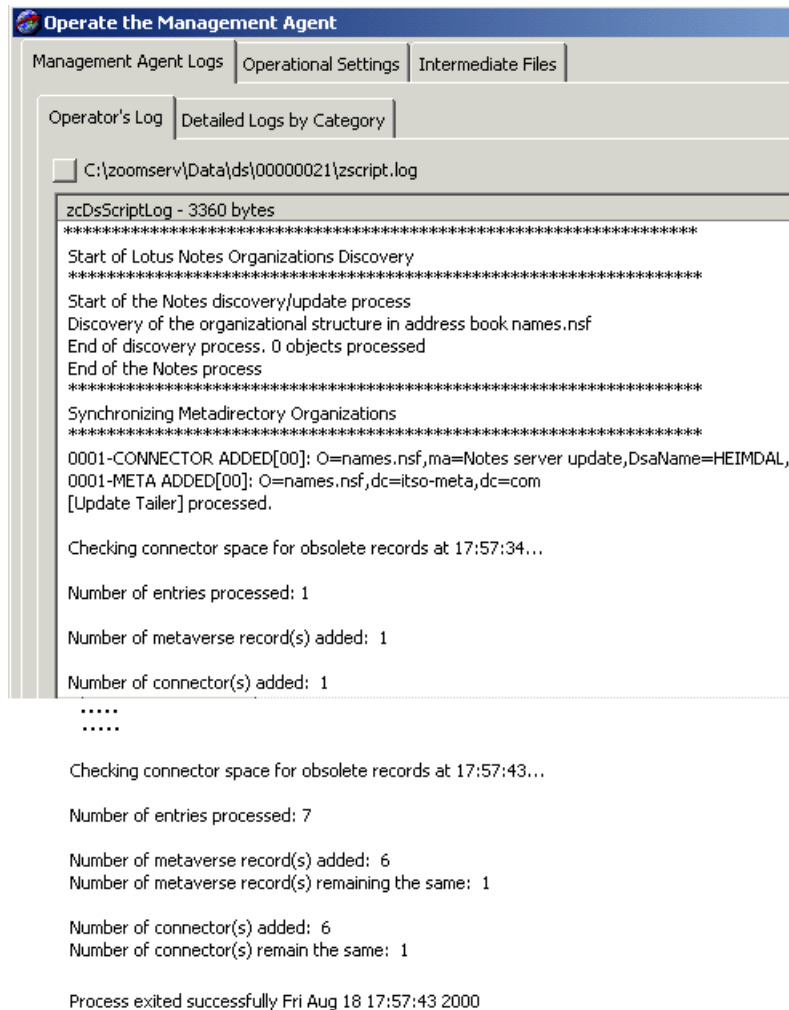


Figure 62. Management agent log

The Lotus Notes management agent updates the metadirectory and also adds a link of each entry below the management agent. These are shown in Figure 63 on page 152 and Figure 64 on page 152, respectively.

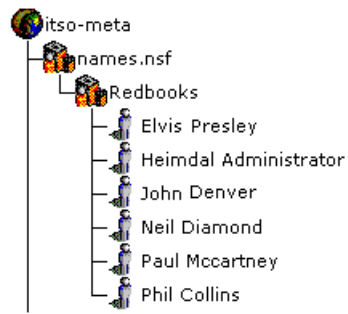


Figure 63. Updating the metadirectory

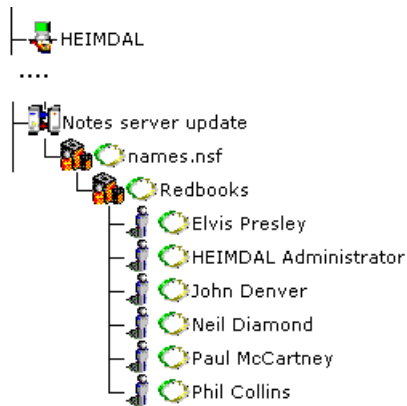


Figure 64. Adding links

You use a browser when updating the metadirectory. The following three figures show the HTTP logon screen, the password challenge and the successful authorization.

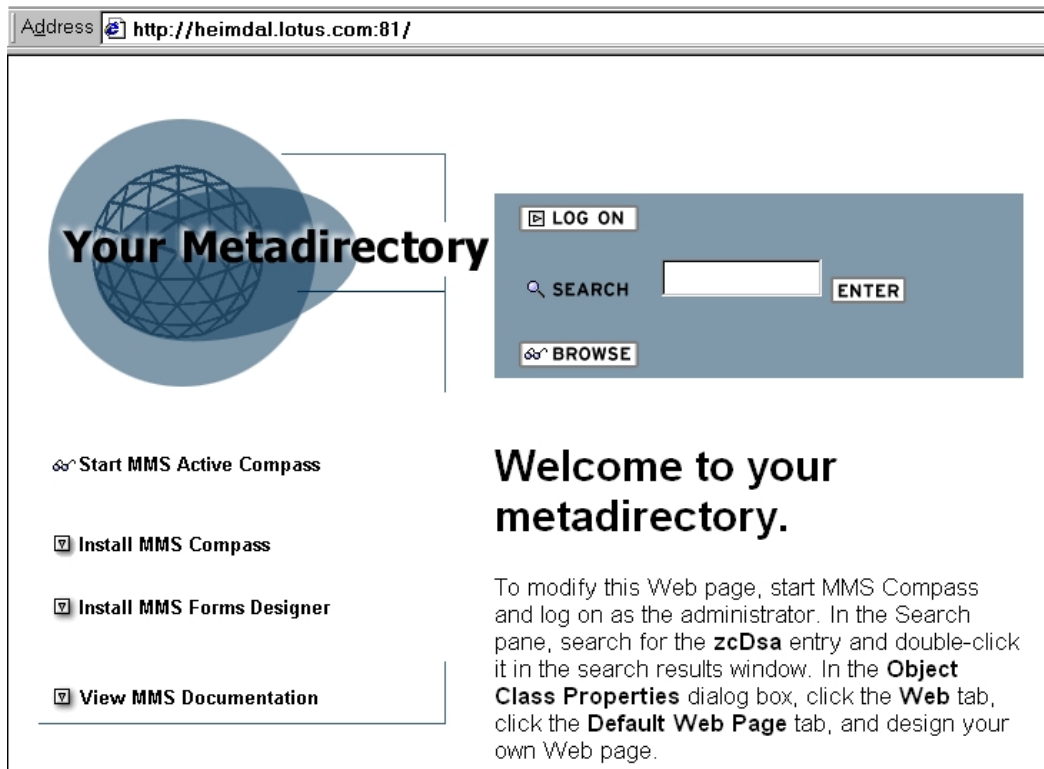


Figure 65. HTTP login screen



Figure 66. Password prompt

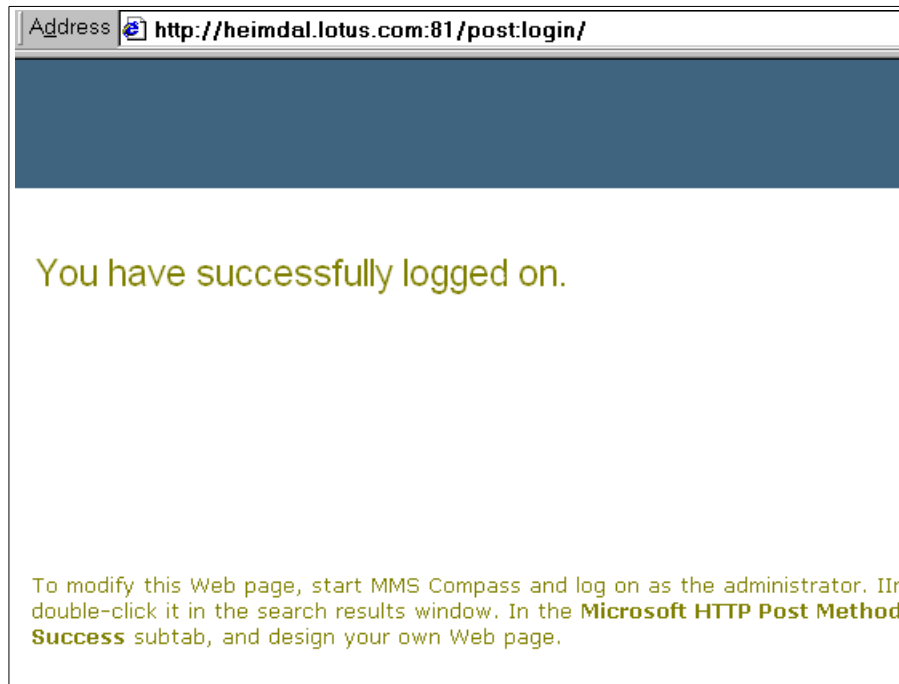


Figure 67. A successful logon

Figure 68 is an example of updating an entry in the metadirectory, and Figure 69 on page 155 confirms that the update was successful.

<p>Move to <b>Redbooks</b></p> <p><b>MODIFY</b></p>		<p>Office Phone <input type="text"/></p> <p>Office Fax <input type="text"/></p> <p>Cellular Phone <input type="text" value="0800 666"/></p> <p>Pager Number <input type="text" value="444 555"/></p> <p>Urgent Calls To <input type="text"/></p>
<p>Owned by <b>Phil Collins:</b></p>	<p><b>OFFICE</b></p>	<p>Office Location <input type="text"/></p> <p>Street Address <input type="text" value="2 Both sides of the story street"/></p>

Figure 68. Updating a metadirectory entry

The screenshot shows a web interface with a light blue background. At the top left, the text "Search for:" is in bold. Below it is a text input field and a "Search" button. To the right of the search area, the message "Your modification was successful." is displayed in a large, bold, black font. Below the search area, the text "Move to Redbooks" is shown, with "Redbooks" underlined. To the left of this text is a button labeled "MODIFY". Below the "MODIFY" button, the text "Owned by Phil Collins:" is displayed. To the right of the "MODIFY" button, there is a paragraph of text: "To modify this Web page, start MMS Compass and log on to search for the **Modify** object and double-click it in the **search** **HTTP Post Method** dialog box, click the **Web Responses** design your own Web page."

**Search for:**

**Search**

**Move to**  
**Redbooks**

**MODIFY**

**Owned by**  
**Phil**  
**Collins:**

To modify this Web page, start MMS Compass and log on to search for the **Modify** object and double-click it in the **search** **HTTP Post Method** dialog box, click the **Web Responses** design your own Web page.

Figure 69. Confirmation of update

We tested the following scenarios and the results were consistent with the attribute flow configuration rules:

- Updating the mobile and pager numbers in the Domino Directory, which updated the metadirectory.
- Updating the mobile and pager numbers in the MMS metadirectory, which did not update the Domino Directory. Instead, the Domino Directory values were added back into the metadirectory.
- Updating the mobile and pager numbers in both directories, which used the new values of the Domino Directory to populate the MMS metadirectory.
- Updating the company and manager name in the MMS directory, which updated the Domino Directory.
- Updating the company and manager name in the Domino Directory, which did not update the MMS metadirectory. Instead, the Domino Directory values were changed back to the metadirectory values.

- Updating the company and manager name in both directories, which used the new values of the MMS metadirectory to overwrite the Domino Directory values.

## **7.2.7 Others**

Other related products that you may encounter include the following:

### **7.2.7.1 Apache Web server**

The Apache Web server is a free, open-source Web server that is widely used. The standard Apache distribution does not include any options for authenticating against an LDAP directory, but there are numerous add-on modules that do provide that functionality. See <http://modules.apache.org> for an extensive list of available modules. A Domino server running the LDAP service should be usable by any of those modules.

### **7.2.7.2 Entrust PKI**

One common use for a directory server is to store the certificates and public keys for a Public Key Infrastructure (PKI). Lotus and Entrust have jointly developed a product, the Lotus Domino Administrative Toolset for Entrust PKI, which synchronizes administration efforts between a Domino Directory and an Entrust PKI environment.



---

## Appendix A. Industry groups

Here are some of the industry groups that are working towards directory-based standardization.

---

### A.1 The Open Group (TOG)

As part of the Open group, the Directory Program Group provides a neutral forum in which customers, vendors and service providers can come together to help bring about interoperability of directories and directory applications.

Using Dirconnect directory interoperability testing, the Open group evolves core requirements for Directory servers' interoperability with LDAP clients.

Web site: <http://www.opengroup.org>

---

### A.2 Directory Interoperability Forum (DIF)

**Note:** This forum has now been incorporated into The Open Group (TOG). See its description above.

This forum was formed to accelerate the evolution and adoption of open directory-based applications. The forum membership includes directory customers, vendors and independent software vendors.

They are interested in advancing open directories based on the LDAP standards, to make directories more usable. This will help ensure that any application written to utilize a directory will be able to run with any directory, regardless of the supplier. This will also make it easy for software developers to create these applications.

Web site: <http://www.directoryforum.org>

---

### A.3 Distributed Management Task Force (DMTF)

This organization is leading the development, adoption, and unification of management standards and initiatives for desktop, enterprise, and internet environments.

By working with key technology vendors and affiliated standards groups, they enable a more integrated, cost-effective, and less crisis-driven approach to management through interoperable management solutions.

They are also working on directory-enabled networks (DEN), where they are working to enable autoconfiguring networks, services and applications. Their aim is to provide a cross-domain solution that will enable interoperability, data sharing and transparency of data source through a common information model (CIM). The aim with the CIM is to have it independent from any protocol, API or repository technology.

Web site: <http://www.dmtf.org>

---

#### **A.4 Directory Services Markup Language (DSML)**

DSML provides a means for representing directory structural information as an XML document. The intent is to allow XML-based enterprise applications to leverage profile and resource information from a directory in their native environment.

DSML allows XML and directories to work together and provide a common ground for all XML-based applications to make better use of directories. The principal goal is to ensure that directories are able to make a growing breed of XML-based applications directory-aware.

Web site: <http://www.dsml.org>

---

## Appendix B. LDAP and X.500 Standards

X.500 is the OSI Directory Standard defined by ISO (<http://www.iso.ch>) and ITU (<http://www.itu.ch>), and is described in the following nine documents, most of which have been available in four editions - 1988, 1993, 1997 and 2001.

ITU-T	ISO	Title
X.500	9594-1	Overview of concepts, models and services
X.501	9594-2	Models
X.509	9594-8	Authentication framework
X.511	9594-3	Abstract service definition
X.518	9594-4	Procedures for distributed operation
X.519	9594-5	Protocol specifications
X.520	9594-6	Selected attribute types
X.521	9594-7	Selected object classes
X.525	9594-9	Replication

The official versions of the ITU documents are available online and require payment. Some "unofficial" versions - often representing latest drafts - are available for download from <ftp://ftp.bull.com/pub/OSIdirectory/>. A useful information site for X.500 can be found at <http://www.nexor.com>.

---

### B.1 LDAPv3

LDAP is not yet a formal IETF "standard", but it is described in a number of RFCs (Request For Comments), which have in many cases informal standard status, and Internet-Drafts, which represent work in progress towards RFC status and generally expire after six months unless a new version is made or it is submitted for RFC status. Most of the documents have been generated by the two principal working groups focused on LDAP work (LDAPEXT and LDUP; see section 2.5.1), as well as individual submissions for areas which fall outside the technical scope of the working groups. In addition, other IETF working groups, such as PKIX, have dependencies on directory and have generated their own LDAP-related work.

Original documents may be found at either <http://www.ietf.org/rfc/> or <http://www.ietf.org/internet-drafts/>

Let us now describe some RFCs and Internet Drafts.

---

## B.2 LDAPv3 RFCs

Here are descriptions of some relevant LDAP RFCs.

### B.2.1 Core Specifications

The LDAPv3 core specifications produced by the IETF were approved by the IESG (Internet Engineering Steering Group) in December 1997 and are contained in RFC 2251-2256.

<i>Title</i>	<b>Lightweight Directory Access Protocol (v3)</b>
<i>RFC</i>	<b>2251</b> <i>[replaces RFC 1777]</i>
<i>Description</i>	The protocol described in this document is designed to provide access to directories supporting the X.500 models, while not incurring the resource requirements of the X.500 Directory Access Protocol (DAP). This protocol is specifically targeted at management applications and browser applications that provide read/write interactive access to directories. When used with a directory supporting the X.500 protocols, it is intended to be a complement to the X.500 DAP.

<i>Title</i>	<b>LDAPv3 Attribute Syntax Definitions</b>
<i>RFC</i>	<b>2252</b> <i>[replaces RFC 1778]</i>
<i>Description</i>	The LDAP requires that the contents of AttributeValue fields in protocol elements be octet strings. This document defines a set of syntaxes for LDAPv3, and the rules by which attribute values of these syntaxes are represented as octet strings for transmission in the LDAP protocol. The syntaxes defined in this document are referenced by this and other documents that define attribute types. This document also defines the set of attribute types which LDAP servers should support.

<i>Title</i>	<b>UTF-8 String Representation of Distinguished Names</b>
<i>RFC</i>	<b>2253</b> <i>[replaces RFC 1779]</i>
<i>Description</i>	The X.500 Directory uses distinguished names as the primary keys to entries in the directory. Distinguished Names are encoded in ASN.1 in the X.500 Directory protocols. In the LDAP, a string representation of distinguished names is transferred. This specification defines the string format for representing names, which is designed to give a clean representation of commonly used distinguished names, while being able to represent any distinguished name.

<i>Title</i>	<b>The String Representation of LDAP Search Filters</b>
<i>RFC</i>	<b>2254</b> <i>[replaces RFC 1960]</i>

<i>Description</i>	The LDAP defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form. This document defines a human-readable string format for representing LDAP search filters. This document replaces RFC 1960, extending the string LDAP filter definition to include support for LDAPv3 extended match filters.
--------------------	--

<i>Title</i>	<b>The LDAP URL Format</b>
<i>RFC</i>	<b>2255</b> <i>[replaces RFC 1959]</i>
<i>Description</i>	This document describes a format for an LDAP Uniform Resource Locator, and describes an LDAP search operation performed to retrieve information from an LDAP directory. It updates the LDAP URL format for LDAPv3. This document also defines a second URL scheme prefix for LDAP running over the TLS protocol.

<i>Title</i>	<b>A Summary of the X.500(96) User Schema for use with LDAPv3</b>
<i>RFC</i>	<b>2256</b>
<i>Description</i>	This document provides an overview of the attribute types and object classes defined by the ISO and ITU-T committees in the X.500 documents, in particular those intended for use by directory clients. This is the most widely used schema for LDAP/X.500 directories, and many other schema definitions for white pages objects use it as a basis. This document does not cover attributes used for the administration of X.500 directory servers, nor does it include attributes defined by other ISO/ITU-T documents.

## B.2.2 Extended Core RFCs

<i>Title</i>	<b>Authentication Methods for LDAP</b>
<i>RFC</i>	<b>2829</b>
<i>Description</i>	This document specifies particular combinations of SASL mechanisms and extensions which are required and recommended in LDAP implementations.

<i>Title</i>	<b>LDAPv3: Extension for Transport Layer Security</b>
<i>RFCe</i>	<b>2830</b>
<i>Description</i>	This document defines the "Start Transport Layer Security (TLS) Operation" for LDAP. This operation provides for TLS establishment in an LDAP association and is defined in terms of an LDAP extended operation.

### B.2.3 Other Related RFCs

<i>Title</i>	<b>The LDAP Application Program Interface</b>
<i>RFC</i>	<b>1823</b>
<i>Description</i>	This document defines a C language application program interface to LDAP, which is designed to be powerful, yet simple to use. It defines compatible synchronous and asynchronous interfaces to LDAP to suit a wide variety of applications. This document gives a brief overview of the LDAP model, then an overview of how the API is used by an application program to obtain LDAP information. The API calls are described in detail, followed by an appendix that provides some example code demonstrating the use of the API.

<i>Title</i>	<b>Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers</b>
<i>RFC</i>	<b>2079</b>
<i>Description</i>	URLs are being widely used to specify the location of Internet resources. There is an urgent need to be able to include URLs in directories that conform to the LDAP and X.500 information models, and a desire to include other types of URIs as they are defined. A number of independent groups are already experimenting with the inclusion of URLs in LDAP and X.500 directories. This document builds on the experimentation to date and defines a new attribute type and an auxiliary object class to allow URIs, including URLs, to be stored in directory entries in a standard way.

<i>Title</i>	<b>Use of an X.500/LDAP directory to support MIXER address mapping</b>
<i>RFC</i>	<b>2164</b>
<i>Description</i>	MIXER (RFC 2156) defines an algorithm for use of a set of global mapping between X.400 and RFC 822 addresses. This specification defines how to represent and maintain these mappings (MIXER Conformant Global Address Mappings of MCGAMs) in an X.500 or LDAP directory. Mechanisms for representing OR Address and Domain hierarchies within the DIT. These techniques are used to define two independent subtrees in the DIT, which contain the mapping information.

<i>Title</i>	<b>A Common Schema for the Internet White Pages Service</b>
<i>RFC</i>	<b>2218</b>

<i>Description</i>	This IETF Integrated Directory Services(IDS) Working Group proposes a standard specification for a simple Internet White Pages service by defining a common schema for use by the various White Pages servers. This schema is independent of specific implementations of the White Pages service. This document specifies the minimum set of core attributes of a White Pages entry for an individual and describes how new objects with those attributes can be defined and published. It does not describe how to represent other objects in the White Pages service. Further, it does not address the search sort expectations within a particular service.
--------------------	--

<i>Title</i>	<b>Simple Authentication and Security Layer (SASL)</b>
<i>RFC</i>	<b>2222</b>
<i>Description</i>	This document describes a method for adding authentication support to connection-based protocols. To use this specification, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating protection of subsequent protocol interactions. If its use is negotiated, a security layer is inserted between the protocol and the connection. This document describes how a protocol specifies such a command, defines several mechanisms for use by the command, and defines the protocol used for carrying a negotiated security layer over the connection.

<i>Title</i>	<b>Using Domains in LDAP/X.500 Distinguished Names</b>
<i>RFC</i>	<b>2247</b>
<i>Description</i>	LDAP uses X.500-compatible distinguished names for providing unique identification of entries. This document defines an algorithm by which a name registered with the Internet Domain Name Service can be represented as an LDAP distinguished name.

<i>Title</i>	<b>An Approach for Using LDAP as a Network Information Service</b>
<i>RFC</i>	<b>2307</b>
<i>Description</i>	This document describes an experimental mechanism for mapping entities related to TCP/IP and the UNIX system into X.500 entries so that they may be resolved with the LDAP. A set of attribute types and object classes are proposed, along with specific guidelines for interpreting them. The intention is to assist the deployment of LDAP as an organizational nameservice. No proposed solutions are intended as standards for the Internet. Rather, it is hoped that a general consensus will emerge as to the appropriate solution to such problems, leading eventually to the adoption of standards. The proposed mechanism has already been implemented with some success.

<i>Title</i>	<b>Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2</b>
--------------	--

<i>RFC</i>	<b>2559</b>
<i>Description</i>	The protocol described in this document is designed to satisfy some of the operational requirements within the Internet X.509 PKI. Specifically, this document addresses requirements to provide access to PKI repositories for the purposes of retrieving PKI information and managing that same information. The mechanism described in this document is based on the LDAPv2, defined in RFC 1777, defining a profile of that protocol for use within the PKIX and updates encodings for certificates and revocation lists from RFC 1778. Additional mechanisms addressing PKIX operational requirements are specified in separate documents.

<i>Title</i>	<b>Internet X.509 Public Key Infrastructure LDAPv2 Schema</b>
<i>RFC</i>	<b>2587</b>
<i>Description</i>	The schema defined in this document is a minimal schema to support PKIX in an LDAPv2 environment, as defined in RFC 2559. Only PKIX-specific components are specified here. LDAP servers, acting as PKIX repositories should support the auxiliary object classes defined in this specification and integrate this schema specification with the generic and other application-specific schemas as appropriate, depending on the services to be supplied by that server.

<i>Title</i>	<b>Extensions for Dynamic Directory Services</b>
<i>RFC</i>	<b>2589</b>
<i>Description</i>	LDAP supports lightweight access to static directory services, allowing relatively fast search and update access. Static directory services store information about people that persists in its accuracy and value over a long period of time. Dynamic directory services are different in that they store information about people that only persists in its accuracy and value while people are online. Though the protocol operations and attributes used by dynamic directory services are similar to the ones used for static directory services, clients that are bound to a dynamic directory service need to periodically refresh their presence at the server to keep directory entries from getting stale in the presence of client application crashes. A flow control mechanism from the server is also described that allows a server to inform clients how often they should refresh their presence.

<i>Title</i>	<b>Use of Language Codes in LDAP</b>
<i>RFC</i>	<b>2596</b>



<i>Description</i>	LDAP provides a means for clients to interrogate and modify information stored in a distributed directory system. The information in the directory is maintained as attributes of entries. Most of these attributes have syntaxes which are human-readable strings, and it is desirable to be able to indicate the natural language associated with attribute values. This document describes how language codes are carried in LDAP and are to be interpreted by LDAP servers. All implementations MUST be prepared to accept language codes in the LDAP protocols. Servers may or may not be capable of storing attributes with language codes in the directory.
--------------------	--

<i>Title</i>	<b>Signed Directory Operations Using S/MIME</b>
<i>RFC</i>	<b>2649</b>
<i>Description</i>	This document defines an LDAPv3 based mechanism for signing directory operations in order to create a secure journal of changes that have been made to each directory entry. Both client and server based signatures are supported. An object class for subsequent retrieval are 'journal entries' is also defined. This document specifies LDAPv3 controls that enable this functionality. It also defines an LDAPv3 schema that allows for subsequent browsing of the journal information.

<i>Title</i>	<b>LDAPv2 Client vs. the Index Mesh</b>
<i>Author</i>	<i>Roland Hedberg</i>
<i>RFC</i>	<b>2657</b>
<i>Description</i>	LDAPv2 clients as implemented according to RFC 1777 have no notion of referral. The integration between such a client and an Index Mesh, as defined by the Common Indexing Protocol, heavily depends on referrals and therefore needs to be handled in a special way. This document defines one possible way of doing this.

<i>Title</i>	<b>LDAP Control Extension for Simple Paged Results Manipulation</b>
<i>RFC</i>	<b>2696</b>
<i>Description</i>	This document describes an LDAPv3 control extension for simple paging of search results. This control extension allows a client to control the rate at which an LDAP server returns the results of an LDAP search operation. This control may be useful when the LDAP client has limited resources and may not be able to process the entire result set from a given LDAP query, or when the LDAP client is connected over a low-bandwidth connection. Other operations on the result set are not defined in this extension. This extension is not designed to provide more sophisticated result set management.

<i>Title</i>	<b>Schema for Representing Java Objects in an LDAP Directory</b>
<i>RFC</i>	<b>2713</b>

<i>Description</i>	This document defines the schema for representing Java objects in an LDAP directory. It defines schema elements to represent a Java serialized object, a Java marshalled object, a Java remote object, and a JNDI reference.
--------------------	--

<i>Title</i>	<b>Schema for Representing CORBA Objects in an LDAP Directory</b>
<i>RFC</i>	<b>2714</b>
<i>Description</i>	CORBA is the Common Object Request Broker Architecture defined by the Object Management Group. This document defines the schema for representing CORBA object references in an LDAP directory.

<i>Title</i>	<b>Calendar Attributes for vCard and LDAP</b>
<i>RFC</i>	<b>2739</b>
<i>Description</i>	When scheduling a calendar entity, such as an event, it is a prerequisite that an organizer has the calendar address of each attendee that will be invited to the event. Additionally, access to an attendee's current "busy time" provides an a priori indication of whether the attendee will be free to participate in the event. In order to meet these challenges, a calendar user agent (CUA) needs a mechanism to locate individual user's calendar and free/busy time. This memo defines three mechanisms for obtaining a URI to a user's calendar and free/busy time. These include: manual transfer of the information; personal data exchange using the vCard format; and directory lookup using the LDAP protocol.

<i>Title</i>	<b>Definition of the inetOrgPerson Object Class</b>
<i>RFC</i>	<b>2798</b>
<i>Description</i>	While the X.500 standards define many useful attribute types [X520] and object classes [X521], they do not define a person object class that meets the requirements found in today's Internet and Intranet directory service deployments. We define a new object class called inetOrgPerson for use in LDAP and X.500 directory services that extends the X.521 standard organizationalPerson class to meet these needs.

<i>Title</i>	<b>Access Control Requirements for LDAP</b>
<i>RFC</i>	<b>2820</b>
<i>Description</i>	This document describes the fundamental requirements of an access control list (ACL) model for the LDAP directory service. It is intended to be a gathering place for access control requirements needed to provide authorized access to and interoperability between directories.

<i>Title</i>	<b>Using Digest Authentication as a SASL Mechanism</b>
--------------	--

<i>RFC</i>	<b>2831</b>
<i>Description</i>	This specification defines how HTTP Digest Authentication can be used as a SASL [RFC 2222] mechanism for any protocol that has a SASL profile. It is intended both as an improvement over CRAM-MD5 [RFC 2195] and as a convenient way to support a single authentication mechanism for web, mail, LDAP, and other protocols.

<i>Title</i>	<b>LDAP Control Extension for Server Side Sorting of Search Results</b>
<i>RFC</i>	<b>2891</b>
<i>Description</i>	This document describes two LDAPv3 control extensions for server side sorting of search results. These controls allows a client to specify the attribute types and matching rules a server should use when returning the results to an LDAP search request. The controls may be useful when the LDAP client has limited functionality or for some other reason cannot sort the results but still needs them sorted. Other permissible controls on search operations are not defined in this extension. The sort controls allow a server to return a result code for the sorting of the results that is independent of the result code returned for the search operation.

### B.3 Internet Draft -- LDAP Extensions

The LDATEXT, LDUP and other working groups define and standardize extensions to the LDAPv3 protocol as well as extensions to the use of LDAP on the Internet. These include the following areas:

#### B.3.1 Applicability and Review of LDAPv3

<i>Title</i>	<b>Lightweight Directory Access Protocol (v3): Applicability Statement</b>
<i>I-D</i>	<i>draft-hodges-ldapv3-as (v0)</i>
<i>Description</i>	The specification for LDAPv3 nominally comprises eight separate RFCs which were issued in two distinct subsets at separate times (RFCs 2251..2256 first, then RFCs 2229 and 2830 following later), but this has never been formally stated. Additionally, RFCs 2251-2256 each are embellished with an "IESG Note" warning implementors and deployers of potential interoperability problems due to the lack of a specification of mandatory-to-implement authentication mechanism(s). This document corrects both situations by explicitly specifying the set of RFCs comprising LDAPv3 and rescinding the "IESG Note" due to the specification of mandatory-to-implement authentication mechanisms in RFC 2829.

<i>Title</i>	<b>The String Representation of LDAP Search Filters</b>
<i>I-D</i>	<i>draft-smith-ldapv3-filter-update (v0)</i>

<i>Description</i>	LDAP defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form. This document defines a human-readable string format for representing the full range of possible LDAPv3 search filters, including extended match filters. This document replaces RFC 2254.
--------------------	---

<i>Title</i>	<b>The LDAP URL Format</b>
<i>I-D</i>	<i>draft-smith-ldapv3-url-update (v0)</i>
<i>Description</i>	LDAP is defined in RFCs 2251-3. This document describes a format for an LDAP UniformResource Locator. The format describes an LDAP search operation to perform to retrieve information from an LDAP directory, or, in the context of an LDAPv3 referral or reference, the format describes a service where an LDAP operation may be progressed. This document specifies the LDAP URL format for LDAPv3 and clarifies how LDAP URLs are resolved. This document also defines an extension mechanism for LDAP URLs, so that future documents can extend their functionality, for example, to provide access to new LDAPv3 extensions as they are defined. This document replaces RFC 2255.

<i>Title</i>	<b>LDAPv3bis Suggestions: Lightweight Directory Access Protocol (v3)</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2251 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to "Lightweight Directory Access Protocol (v3)" [RFC2251]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: Attribute Syntax Definitions</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2252 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to " Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions" [RFC2252]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: UTF-8 String Representation of Distinguished Names</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2253 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names" [RFC2253]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: The String Representation of LDAP Search Filters</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2254 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to "The String Representation of LDAP Search Filters" [RFC 2254]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: The LDAP URL Format</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2255 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to "The LDAP URL Format" [RFC 2255]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: Summary of the X.500(96) User Schema for use with LDAPv3</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2256 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to "A Summary of the X.500(96) User Schema for use with LDAPv3" [RFC 2256]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: Authentication Methods for LDAP</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2829 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to "Authentication Methods for LDAP" [RFC2829]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

<i>Title</i>	<b>LDAPv3bis Suggestions: Extension for Transport Layer Security</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-rfc2830 (v0)</i>
<i>Description</i>	This Internet Draft suggests a number of updates to the "Lightweight Directory Access Protocol: Extension for Transport Layer Security" [RFC 2830]. This document is not intended to be published as an RFC but used to identify LDAPv3bis work items.

### B.3.2 LDAP Controls & Operations

<i>Title</i>	<b>LDAP Extension Style Guide</b>
<i>I-D</i>	<i>draft-greenblatt-ldapextstyle (v0)</i>

<i>Description</i>	LDAPv3 provides a base set of services. Additionally, LDAP provides several mechanisms by which the base set of services may be enhanced to provide additional services. This document describes the different ways that LDAP may be enhanced, and how developers can decide which enhancement mechanism is best suited for their environment. It also discusses the positives and negatives for each LDAP enhancement mechanism
--------------------	--

<i>Title</i>	<b>Result Message for LDAP Controls</b>
<i>I-D</i>	<i>draft-armijo-ldap-control-error (v0)</i>
<i>Description</i>	LDAPv3 allows for the extension of the protocol through the use of controls. These controls allow existing operations to be enhanced to provide additional functionality for directory operations. Complex controls are being established that are bringing up error conditions not anticipated in the LDAPv3 specifications. The purpose of this draft is to create new result codes specific to LDAP controls and to define guidelines for the use of these result codes.

<i>Title</i>	<b>Tree Delete Control</b>
<i>I-D</i>	<i>draft-armijo-ldap-treedelelete (v2)</i>
<i>Description</i>	This document defines an LDAPv3 control that deletes an entire subtree of a container entry. This control extends the scope of the LDAPv3 delete operation as defined in RFC 2251. This control is beneficial in extending the functionality of the LDAP protocol and may be useful in administration in an LDAP environment.

<i>Title</i>	<b>Simple Operations on Subtrees (for LDAP)</b>
<i>I-D</i>	<i>draft-greenblatt-ldapext-sos (v1)</i>

<i>Description</i>	<p>This draft defines several new LDAP extensions, which are operations that can manipulate an entire portion of Directory Information Tree (DIT) at once. This draft does not presume any specific DIT structure or schema modifications.</p> <p>Here are some requirements that for building real world LDAP applications that try to operate on an entire subtree.</p> <ul style="list-style-type: none"> <li>- Provide user feedback as to the progress of the long lived operation. In many scenarios, a subtree operation (e.g. subtree copy) may take a long period of time (many hours for large subtrees). It is essential to have a progress bar move across the screen as the entries are deleted.</li> <li>- As the delete subtree crosses containers into other LDAP servers, additional authentication credentials may be required to be retrieved from the LDAP client, in order to allow the operation to proceed.</li> <li>- If the authenticated user has access to only a portion of the sub-tree to be deleted, it should be possible for the part of the sub-tree that is possible to delete, to in fact be deleted. It should also be possible to submit the operation in such a way that no entries from the subtree are deleted unless it is possible to delete all entries from the subtree.</li> <li>- The list of entries that has been deleted by the operation should be returned to the client. An incremental list of deleted entries could be returned with the progress indication above.</li> <li>- It should be possible to "cancel" the delete subtree operation, just as the long lived Search operation can be "abandoned".</li> <li>- It should be possible to delete only certain types of entries from the subtree. For example, delete all printer objects from the subtree.</li> </ul> <p>Note that this current draft does not necessarily address all of the requirements above.</p>
--------------------	---

<i>Title</i>	<b>EntrySelection Control for LDAP Modify and Delete Operations on Multiple Entries</b>
<i>I-D</i>	<i>draft-haripriya-ldapext-entryselect (v1)</i>

<i>Description</i>	<p>This document defines an LDAPv3 control that can select multiple entries in a subtree of a container entry for modification or deletion. This control extends the scope of the LDAPv3 modify and delete operations as defined in [RFC 2251]. This control is useful for modifying or deleting multiple entries on the basis of a single selection criterion. This may be useful for maintenance of an LDAP directory having a large number of objects.</p> <p>Example of Usage - This control can be used by client applications who have the need to modify or delete a large number of entries in an LDAP directory based on a selection criterion. One example of such a usage is when two departments in an organization merge. In this case the "department" name or number given to a number of employees need to change, and all the employees in a given department are to be assigned the new "department". Here the EntrySelection control can be used to select the entries to be modified based on the "department" value, and the modify operation can change the "department" value for all the selected entries to the given value</p> <p>The EntrySelection control is useful when a large number of entries have to be modified or deleted, because what can be achieved in 1 LDAP client operation with the EntrySelection control will take a minimum of 1 + n LDAP operations (1 search, n modifies) otherwise. This will save a lot of time and bandwidth, and hence very useful in situations where the clients are connected over low bandwidth and/or high latency links. Also low-end clients which cannot handle a large number of objects, can use this feature. This also prevents cache pollution or false caching, where a large number of search results are returned only to be immediately modified or deleted, thus invalidating cached information for those results.</p>
--------------------	---

<i>Title</i>	<b>LDAP Control for a Duplicate Entry Representation of Search Results</b>
<i>I-D</i>	<i>draft-ietf-ldapext-ldapv3-dupent (v4)</i>
<i>Description</i>	<p>This document describes a Duplicate Entry Representation control extension for the LDAP Search operation. By using the control with an LDAP search, a client requests that the server return separate entries for each value held in the specified attributes. For instance, if a specified attribute of an entry holds multiple values, the search operation will return multiple instances of that entry, each instance holding a separate single value in that attribute.</p>

<i>Title</i>	<b>Extensible Match Rule to Dereference Pointers</b>
<i>I-D</i>	<i>draft-moats-ldap-dereference-match (v2)</i>



<i>Description</i>	<p>This document defines a LDAPv3 extensible matching rule that allows a server to dereference pointers stored in an object's attribute and apply a LDAPv3 search filter to the resulting objects. This rule allows schema definitions to capture richer association models without requiring extra protocol exchanges or special client code.</p> <p>When mapping rich information models, it sometimes becomes necessary to use DN pointers to show relationships between objects in the schema. An example is the information model and resulting core schema that is currently being proposed by the policy working group. To maintain client efficiency, it is desirable to define an extensible match rule that follows DN pointers as part of a query.</p>
--------------------	---

<i>Title</i>	<b>Returning Matched Values with LDAPv3</b>
<i>I-D</i>	<i>draft-ietf-ldapext-matchedval (v3)</i>
<i>Description</i>	<p>This document describes a control for the LDAPv3 that is used to return a subset of attribute values from an entry, specifically, only those values that contributed to the search filter evaluating to TRUE. Without support for this control, a client must retrieve all of an attribute's values and search for specific values locally.</p>

<i>Title</i>	<b>LDAP Controls for Reply Signatures</b>
<i>I-D</i>	<i>draft-salzr-ldap-repsig (v0)</i>
<i>Description</i>	<p>In many environments the final step of certificate issuance is publishing the certificate to a repository. Unfortunately, there is no way for a Certification Authority (CA) to have a secure application-level acknowledgement that the proper repository did, in fact, receive the certificate. This issue is of greater concern when considering the publication of Certificate Revocation Lists (CRLs) -- if an adversary manages to interpose itself between the CA and its intended repository, then clients could end up relying on outdated revocation lists. This document defines a set of controls so that an LDAP client, such as a CA, can receive a cryptographically secure acknowledgement that an LDAP server has received a request, and that the integrity of the server's reply has not been compromised. Whenever possible, the definitions here use mechanisms and datatypes defined by the IETF PKIX working group. This document references RFC 2459. Knowledge of the RFC is required for proper implementation of this document, although it should be possible to understand this document without much knowledge of that RFC. It is expected that future versions of this document will reference 2459's successor(s).</p>

<i>Title</i>	<b>LDAPv3 Result Codes: Definitions and Appropriate Use</b>
<i>I-D</i>	<i>draft-just-ldapv3-rescodes (v2)</i>

<i>Description</i>	The purpose of this document is to describe, in some detail, the meaning and use of the result codes used with the LDAPv3 protocol. Of particular importance are the error codes, which represent the majority of the result codes. This document provides definitions for each result code, and outlines the expected behaviour of the various operations with respect to how result codes and in particular, error conditions should be handled and which specific error code should be returned. It is hoped that this document will facilitate interoperability between clients and servers and the development of intelligent LDAP clients capable of acting upon the results received from the server.
--------------------	--

<i>Title</i>	<b>LDAPv3: Grouping of Related Operations</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-grouping (v0)</i>
<i>Description</i>	This document provides a general mechanisms for grouping related LDAP operations, which may be used to support replication, proxies, and higher level operations such as transactions. This document describes a set of LDAP extended operations and other protocol and schema elements to support grouping of related operations.

<i>Title</i>	<b>LDAPv3 Transactions</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-txn (v0)</i>
<i>Description</i>	LDAP update operations have atomic properties upon individual entries. However, it is often desirable to update two or more entries as one atomic action, a transaction. Transactions are necessary to support a number of applications including resource provisioning and information replication. This document defines an LDAP extension to support transactions.

<i>Title</i>	<b>LDAPv3: All Operational Attributes</b>
<i>I-D</i>	<i>draft-zeilenga-ldapv3bis-opattr (v0)</i>
<i>Description</i>	X.500 provides a mechanism for clients to request all operational attributes be returned with entries provided in response to a search operation. LDAP [RFC2251] does not provide a similar mechanism to clients to request the return of operational attributes. The lack of such a mechanisms hinders discovery of operational attributes present in an entry This document defines a simple mechanism which clients may use to request all operation attributes. This document updates RFC 2251 as detailed below.

### B.3.3 CLDAP

<i>Title</i>	<b>Connection-less Lightweight Directory Access Protocol</b>
<i>I-D</i>	<i>draft-ietf-ldapext-cldap (v0)</i>

<i>Description</i>	This memo describes modifications to LDAPv3 to allow transport of a subset of the LDAP protocol over connection-less transport. The case of UDP/IP is covered in detail in this memo but other transport layers are possible.
--------------------	---

### B.3.4 Other Extensions

<i>Title</i>	<b>Storing Vendor Information in the LDAP root DSE</b>
<i>I-D</i>	<i>draft-mmoredith-rootdse-vendor-info (v2)</i>
<i>Description</i>	This document specifies two LDAP attributes, vendorName and vendorVersion that MAY be included in the root DSE to advertise vendor-specific information. These two attributes supplement the attributes defined in section 3.4 of RFC 2251. The information held in these attributes MAY be used for display and informational purposes and MUST NOT be used for feature advertisement or discovery.

<i>Title</i>	<b>The LDAP Caching model</b>
<i>I-D</i>	<i>draft-natarajan-ldapext-cachedresults (v0)</i>
<i>Description</i>	Seeking entries from a directory is a process involving network resources. It is assumed that a directory is accessed for reading and searching data more than for modification purposes. Under such assumptions, for performance reasons, a mechanism for caching as a proxy which caches all entries is desirable. This document describes a mechanism for caching directory entries. This document also defines one operational attribute and two controls required to be implemented for the caching model.

<i>Title</i>	<b>Administrator Address Attribute</b>
<i>I-D</i>	<i>draft-wahl-ldap-adminaddr (v0)</i>
<i>Description</i>	Organizations running multiple directory servers need an ability for administrators to determine who is responsible for a particular server. This is conceptually similar to the 'sysContact' object of SNMP. The administratorsAddress attribute allows a server administrator to provide the contact information of the responsible party for an LDAP server. This can be used by management clients which are, for example, checking the state of a replication or referral topology, to provide a way for the user of the management client to send email to manager of a particular server.

<i>Title</i>	<b>Extended Partial Response Protocol Enhancement to LDAPv3</b>
<i>I-D</i>	<i>draft-rharrison-ldap-extpartresp (v1)</i>

<i>Description</i>	This document describes the ExtendedPartialResponse, an element of LDAP v3 protocol which allows multiple responses to LDAPv3 extended requests. Extended partial responses are backward compatible with the existing LDAPv3 Extended Operation defined in LDAPv3..
--------------------	---

### B.3.5 Access Control, Authentication & Authorization

<i>Title</i>	<b>Password Policy for LDAP Directories</b>
<i>I-D</i>	<i>draft-behera-ldap-password-policy (v2)</i>
<i>Description</i>	Password policy is a set of rules that controls how passwords are used in LDAP directories. In order to improve the security of LDAP directories and make it difficult for password cracking programs to break into directories, it is desirable to enforce a set of rules on password usage. These rules are made to ensure that users change their passwords periodically, passwords meet construction requirements, the re-use of old password is restricted, and users are locked out after a certain number of failed attempts.

<i>Title</i>	<b>Access Control Model for LDAP</b>
<i>I-D</i>	<i>draft-ietf-ldapext-acl-model (v6)</i>
<i>Description</i>	This document describes the access control list (ACL) model for an LDAP directory service. It includes a description of the model, the LDAP controls, and the extended operations to the LDAP protocol. A separate document defines the corresponding APIs.

<i>Title</i>	<b>X.509 Authentication SASL Mechanism</b>
<i>I-D</i>	<i>draft-ietf-ldapext-x509-sasl (v3)</i>
<i>Description</i>	This document defines a SASL [RFC 2222] authentication mechanism based on X.509 strong authentication, providing two way authentication. This mechanism is only for authentication, and has no effect on the protocol encodings and is not designed to provide integrity or confidentiality services.

<i>Title</i>	<b>Internet X.509 Public Key Infrastructure Additional LDAP Schema for PKIs and PMIs</b>
<i>I-D</i>	<i>draft-ietf-pkix-ldap-schema (v0)</i>

<i>Description</i>	This document describes LDAP schema features in addition to RFC 2587 that are needed to support a Privilege Management Infrastructure and a Public Key Infrastructure. RFC2587 describes some of the subschema applicable to LDAPv2 servers, specifically the public key certificate related attribute types and object classes that MUST or MAY be supported. This document does not revoke any of the contents of RFC2587, but supplements them. RFC2587 is equally applicable to LDAPv3 servers as to LDAPv2 servers and MUST be supported by LDAPv3 servers. Neither RFC2587 nor the user schema for LDAPv3 (RFC2256) nor the attribute syntax definitions for LDAPv3 (RFC2252) describe in detail the matching rules that should be supported by LDAP servers, nor do they describe how attribute value assertions for each matching rule should be encoded in filter items. Finally none of these documents mention attributeCertificates or any schema to support privilege management, since these concepts superseded the publishing of the RFCs.
--------------------	--

<i>Title</i>	<b>An Example of DIGEST-MD5 Authentication within an LDAP server</b>
<i>I-D</i>	<i>draft-wahl-ldap-digest-example (v0)</i>
<i>Description</i>	HTTP Digest Authentication as a SASL mechanism is required to be supported in LDAP servers for password-based authentication (see Authentication Methods for LDAP). This specification describes one approach to implement DIGEST-MD5 authentication in an LDAP server. It does not specify a standard of any kind.

<i>Title</i>	<b>LDAP Authentication Response Control</b>
<i>I-D</i>	<i>draft-weltman-ldapv3-auth-response (v1)</i>
<i>Description</i>	This document defines support for the Authentication Response Control. Controls are an LDAPv3 extension, to allow passing arbitrary control information along with a standard request to a server, and to receive arbitrary information back with a standard result. The Authentication Response Control may be returned by an LDAP server in a bind response to a client authenticating with LDAPv3. The control contains the identity assumed by the client. This is useful when there is a mapping step or other indirection during the bind, so that the client can be told what LDAP identity was granted. Client authentication with certificates is the primary situation where this applies. Also, some SASL authentication mechanisms may not involve the client explicitly providing a DN.

<i>Title</i>	<b>LDAP Proxied Authorization Control</b>
<i>I-D</i>	<i>draft-weltman-ldapv3-proxy (v4)</i>

<i>Description</i>	This document defines support for the Proxied Authorization Control. Controls are an LDAPv3 extension, to allow passing arbitrary control information along with a standard request to a server, and to receive arbitrary information back with a standard result. The Proxied Authorization Control allows a connection with sufficient privileges to assume the identity of another entry for the duration of an LDAP request.
--------------------	--

<i>Title</i>	<b>LDAP Authentication Password Attribute</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-authpasswd (v3)</i>
<i>Description</i>	This document describes schema for storing authentication passwords in an LDAP directory. The document provides schema definitions for authPassword and related schema definitions. The authPassword is intended to be used instead of clear text password storage mechanisms such as userPassword [RFC2256] to support simple bind operations. The attribute may be used to store SASL authentication passwords in entries of a directory.

<i>Title</i>	<b>LDAP Password Modify Extended Operation</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-passwd-exop (v4)</i>
<i>Description</i>	The integration of LDAP and external authentication services has introduced non-DN authentication identities and allowed for non-directory storage of passwords. As such, mechanisms which update the directory, such as Modify operation, cannot be used to change a user's password. This document describes an LDAP extended operation to allow modification of user passwords which is not dependent upon the form of the authentication identity nor the password storage mechanism used.

### B.3.6 Sorting and paged retrieval of search results

<i>Title</i>	<b>LDAP Extensions for Scrolling View Browsing of Search Results</b>
<i>I-D</i>	<i>draft-ietf-ldapext-ldapv3-vlv (v4)</i>
<i>Description</i>	<p>This document describes a Virtual List View control extension for the LDAP Search operation. This control is designed to allow the "virtual list box" feature, common in existing commercial e-mail address book applications, to be supported efficiently by LDAP servers. LDAP servers' inability to support this client feature is a significant impediment to LDAP replacing proprietary protocols in commercial e-mail systems.</p> <p>The control allows a client to specify that the server return for a given LDAP search with associated sort keys, a contiguous subset of the search result set. This subset is specified in terms of offsets into the ordered list, or in terms of a greater than or equal comparison value.</p>

<i>Title</i>	<b>Persistent Search: A Simple LDAP Change Notification Mechanism</b>
<i>Author</i>	Mark Smith, Gordon Good, Tim Howes, Rob Weltman
<i>Description</i>	This document defines two controls that extend the LDAPv3 search operation to provide a simple mechanism by which an LDAP client can receive notification of changes that occur in an LDAP server. The mechanism is designed to be very flexible yet easy for clients and servers to implement.

### B.3.7 Directory-Enabled Networking

<i>Title</i>	<b>LDAP Schema for the DMTF Application CIM v2.1 Model</b>
<i>I-D</i>	<i>draft-moats-dmtf-application-ldap (v1)</i>
<i>Description</i>	This draft presents a LDAPv3 schema for the DMTF CIM Application model. Associations are mapped using a combination of auxiliary classes and DIT structure rules. Where auxiliary classes are used, name form and DIT content rules are specified. (This document is not a product of the DMTF, and represents the view of the authors.)

<i>Title</i>	<b>LDAP Schema for the DMTF Core CIM v2.2 Model</b>
<i>I-D</i>	<i>draft-moats-dmtf-core-ldap (v1)</i>
<i>Description</i>	This draft presents a LDAPv3 schema for the DMTF CIM Core model. Associations are mapped using a combination of auxiliary classes and DIT structure rules. All attribute, object class, and name form OIDs are place holders, and syntax OIDs in definitions have been replaced by names for clarity. Further, structure rule identifiers are place holders and should be replaced as dictated by local implementations. (This document is a product of the DMTF LDAP WG.)

<i>Title</i>	<b>LDAP Schema for the DMTF Device CIM v2.2 Model</b>
<i>I-D</i>	<i>draft-moats-dmtf-device-ldap (v1)</i>
<i>Description</i>	This draft presents a LDAPv3 schema for the DMTF CIM Device model. It builds on the core model presented in <i>draft-moats-dmtf-core-ldap (v1)</i> . Associations are mapped using a combination of auxiliary classes and DIT structure rules. Where auxiliary classes are used, name form and DIT content rules are specified. (This document is not a product of the DMTF, and represents the view of the authors.)

<i>Title</i>	<b>LDAP Schema for the DMTF Network CIM v2.2 Model</b>
<i>I-D</i>	<i>draft-moats-dmtf-network-ldap (v1)</i>

<i>Description</i>	This draft presents a LDAPv3 schema for the DMTF CIM Network model. Associations are mapped using a combination of auxiliary classes and DIT structure rules. Where auxiliary classes are used, name form and DIT content rules are specified. (This document is not a product of the DMTF, and represents the view of the authors.)
--------------------	--

<i>Title</i>	<b>LDAP Schema for the DMTF Physical CIM v2.2 Model</b>
<i>I-D</i>	<i>draft-moats-dmtf-physical-ldap (v1)</i>
<i>Description</i>	This draft presents a LDAPv3 schema for the DMTF CIM Physical model. Associations are mapped using a combination of auxiliary classes and DIT structure rules. Where auxiliary classes are used, name form and DIT content rules are specified. (This document is not a product of the DMTF, and represents the view of the authors.)

<i>Title</i>	<b>LDAP Schema for the DMTF System CIM v2.2 Model</b>
<i>I-D</i>	<i>draft-moats-dmtf-system-ldap (v1)</i>
<i>Description</i>	This draft presents a LDAPv3 schema for the DMTF CIM System model. It builds on the core model presented in <i>draft-moats-dmtf-core-ldap (v1)</i> . Associations are mapped using a combination of auxiliary classes and DIT structure rules. Where auxiliary classes are used, name form and DIT content rules are specified. (This document is not a product of the DMTF, and represents the view of the authors.)

### B.3.8 LDAP APIs

<i>Title</i>	<b>The C LDAP Application Program Interface</b>
<i>I-D</i>	<i>draft-ietf-ldapext-ldap-c-api (v4)</i>
<i>Description</i>	This document defines a C language application program interface to LDAP, and replaces the previous definition of this API, defined in RFC 1823, updating it to include support for features found in LDAPv3, as well as other changes to support information hiding and thread safety.

<i>Title</i>	<b>LDAP C API Concurrency Extensions</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-c-api-concurrency (v0)</i>
<i>Description</i>	This document defines extensions to the LDAP C API to support use in concurrent execution environments. The document describes and defines requirements for multiple concurrency levels: thread safe, session thread safe, and operation thread safe.



<i>Title</i>	<b>LDAP C API Error Reporting Extension</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-c-api-errno (v0)</i>
<i>Description</i>	This document defines a mandatory extension to the LDAP C API to provide error reporting for all API calls. The mechanism is non-intrusive and can, optionally, support concurrent execution environments.

<i>Title</i>	<b>C LDAP API LDERRNO Extension</b>
<i>I-D</i>	<i>draft-smith-ldap-c-api-ext-lderrno (v0)</i>
<i>Description</i>	This document defines an extension to the C LDAP API to support reporting of specific errors for functions in the API that do not provide a way to access detailed information about failures. Three new functions are defined: <code>ldap_get_lderrno()</code> , <code>ldap_set_lderrno()</code> , and <code>ldap_dup_string()</code> .

<i>Title</i>	<b>LDAP C API Virtual List View Extension (VLV)</b>
<i>I-D</i>	<i>draft-smith-ldap-c-api-ext-vlv (v0)</i>
<i>Description</i>	This document defines a virtual list view extension for the LDAP C API to support the LDAP protocol extensions for scrolling view browsing of search results. More specifically, this document defines functions to create virtual list view request controls and to parse virtual list view response controls.

<i>Title</i>	<b>The Java LDAP Application Program Interface</b>
<i>I-D</i>	<i>draft-ietf-ldapext-ldap-java-api (v11)</i>
<i>Description</i>	This document defines a java language application program interface to the LDAP, in the form of a class library. It complements but does not replace the C language API. This version adds support for SASL authentication.

<i>Title</i>	<b>The Java LDAP Application Program Interface Asynchronous Extension</b>
<i>I-D</i>	<i>draft-ietf-ldapext-ldap-java-api-asynch-ext (v5)</i>
<i>Description</i>	This document defines asynchronous extensions to the java language application program interface to LDAP defined in <i>draft-ietf-ldapext-ldap-java-api (v7)</i>

<i>Title</i>	<b>The Java SASL Application Program Interface</b>
<i>I-D</i>	<i>draft-weltman-java-sasl (v3)</i>

<i>Description</i>	This document defines a client-side and a server-side Java language interface for using the Simple Authentication and Security Layer (SASL) mechanisms for adding authentication support to connection-based protocols. The interface promotes sharing of SASL mechanism drivers and security layers between applications using different protocols. It complements but does not replace [SASL], which defines and exemplifies use of the SASL protocol in a language-independent way.
--------------------	--

<i>Title</i>	<b>Java LDAP Controls</b>
<i>I-D</i>	<i>draft-weltman-ldap-java-controls (v4)</i>
<i>Description</i>	This document defines support for the Preferred Language Control, the Server Sorting Control, and the Virtual List Control in the Java LDAP API. Controls are an LDAPv3 extension, to allow passing arbitrary control information along with a standard request to a server, and to receive arbitrary information back with a standard result.

### B.3.9 Synchronization

<i>Title</i>	<b>The LDAP Data Interchange Format (LDIF) - Technical Specification</b>
<i>I-D</i>	<i>draft-good-ldap-ldif (v6)</i>
<i>Description</i>	This document describes a file format suitable for describing directory information or modifications made to directory information. The file format, known as LDIF, for LDAP Data Interchange Format, is typically used to import and export directory information between LDAP-based directory servers, or to describe a set of changes which are to be applied to a directory.

<i>Title</i>	<b>Definition of an Object Class to Hold LDAP Change Records</b>
<i>I-D</i>	<i>draft-good-ldap-changelog (v1)</i>
<i>Description</i>	In order to support more flexible replication methods, it is desirable to specify some manner in which an LDAP client may retrieve a set of changes which have been applied to an LDAP server's database. The client, which may be another LDAP server, may then choose to update its own replicated copy of the data. This document specifies an object class which may be used to represent changes applied to an LDAP server. It also specifies a method for discovering the location of the container object which holds these change records, so that clients and servers have a common rendezvous point for this information.

### B.3.10 Replication (LDUP/LCUP)

<i>Title</i>	<b>Extended Operations for Framing LDAP Operations</b>
--------------	--

<i>I-D</i>	<i>draft-ietf-ldup-framing (v0)</i>
<i>Description</i>	Certain types of LDAP applications can benefit from the ability to specify the beginning and end of a related group of operations. For example, the LDUP multimaster update protocol requires that two servers agree to begin a session to transfer pending replication updates. This document provides a framework for constructing protocols that feature a framed set of related operations. It defines a pair of LDAPv3 extended operations that provide begin-end framing, and a pair of extended operations used to respond the begin-end framing operations. The nature of the actual LDAP operations carried inside these framing operations is not specified in this document.

<i>Title</i>	<b>LDUP Replication Information Model</b>
<i>I-D</i>	<i>draft-ietf-ldup-infomod (v1)</i>
<i>Description</i>	<p><i>draft-merrells-ldup-model (v1)</i> describes the architectural approach to replication of LDAP directory contents. This document describes the information model and schema elements which support LDAP Replication Services which conform to <i>draft-merrells-ldup-model (v1)</i>.</p> <p>Directory schema is extended to provide object classes, subentries, and attributes to describe areas of the namespace which are under common administrative authority, units of replication (ie, subtrees, or partitions of the namespace, which are replicated), servers which hold replicas of various types for the various partitions of the namespace, which namespaces are held on given servers, and the progress of various namespace management and replication operations. Among other things, this knowledge of where directory content is located will provide the basis for dynamic generation of LDAP referrals for clients who can follow them. The controlling framework by which the relationships, types, and health of replicas of the directory content will be defined so that, as much as possible, directory content is itself used to monitor and control the environment.</p> <p>Security information, including access control policy identifiers and information will be treated as directory content by the replication protocols when specified by the LDAPv3 group.</p> <p>The information model will describe required and optional house-keeping duties for compliant systems to implement, such as garbage collection of deleted objects, reconciliation of moved and renamed objects, update sequencing and transaction bracketing of changes, etc.</p>

<i>Title</i>	<b>LDAP Replication Architecture</b>
<i>I-D</i>	<i>draft-ietf-ldup-model (v3)</i>
<i>Description</i>	This architectural document outlines a suite of schema and protocol extensions to LDAPv3 that enables the robust, reliable, server-to-server exchange of directory content and changes.

<i>Title</i>	<b>The LDUP Replication Update Protocol</b>
<i>I-D</i>	<i>draft-ietf-ldup-protocol (v2)</i>
<i>Description</i>	The protocol described in this document is designed to allow one LDAP server to replicate its directory content to another LDAP server. The protocol is designed to be used in a replication configuration where multiple updatable servers are present. Provisions are made in the protocol to carry information that allows the server receiving updates to apply a total ordering to all updates in the replicated system. This total ordering allows all replicas to correctly resolve conflicts that arise when LDAP clients submit changes to different servers that later replicate to one another.

<i>Title</i>	<b>LDAP V3 Replication Requirements</b>
<i>I-D</i>	<i>draft-ietf-ldup-replica-req (v3)</i>
<i>Description</i>	This document discusses the fundamental requirements for replication of data accessible via the LDAPv3 protocol. It is intended to be a gathering place for general replication requirements needed to provide interoperability between informational directories.

<i>Title</i>	<b>LDAP Subentry Schema</b>
<i>I-D</i>	<i>draft-ietf-ldup-subentry (v3)</i>
<i>Description</i>	This document describes an object class called ldapSubEntry which MAY be used to indicate operations and management related entries in the directory, called LDAP Subentries. This version of this document is updated with an assigned OID for the ldapSubEntry object class.

<i>Title</i>	<b>LDUP Update Reconciliation Procedures</b>
<i>I-D</i>	<i>draft-ietf-ldup-urp (v3)</i>
<i>Description</i>	This document describes the procedures used by directory servers to reconcile updates performed by autonomously operating directory servers in a distributed, replicated directory service.

<i>Title</i>	<b>LDAP Client Update Protocol</b>
<i>I-D</i>	<i>draft-natkovich-ldap-lcup (v1)</i>
<i>Description</i>	This document defines the LDAP Client Update Protocol (LCUP). The protocol is intended to allow an LDAP client to synchronize with the content of a directory information tree (DIT) stored by an LDAP server and to be notified about the changes to that content.

<i>Title</i>	<b>LDAP Bulk Update/Replication Protocol</b>
<i>I-D</i>	<i>draft-rharrison-lburp (v2)</i>
<i>Description</i>	<p>The LDAP Bulk Update/Replication Protocol (LBURP) described in this document allows an LDAP client (a genuine client or an LDAP server acting as a client) to perform a bulk update to a replica on an LDAP server. The protocol groups a set of update operations using the LDAP framed protocol requests defined in [FRAMING] to notify the client that the update operations in the framed set are related. The update operations within the framed set are LDAPv3 extended operations each encapsulating a sequence number and one or more LDAPv3 update operations. The sequence number allows the server to process the update operations in the proper order even when they are sent asynchronously by the client, and the update operations can be grouped within the extended request to maximize the efficiency of client-server communication.</p> <p>The protocol may be used to initialize all of the entries in an LDAP replica or to incrementally update the existing entries in an LDAP replica. It is suitable for client utilities that need to efficiently initialize a replica with many entries or efficiently make a substantial set of update changes to a replica. It is also suitable as a protocol for replication between a single master replica and its slave replicas.</p>

### B.3.11 Internet Directory Services

<i>Title</i>	<b>Technical Infrastructure for Swedish Directory Access Gateways (TISDAG)</b>
<i>I-D</i>	<i>draft-daigle-tisdag (v2)</i>
<i>Description</i>	<p>The strength of the TISDAG project's DAG proposal is that it defines the necessary technical infrastructure to provide a single-access-point service for information on Swedish Internet users. The resulting service will provide uniform access for all information -- the same level of access to information (7x24 service), and the same information made available, irrespective of the service provider responsible for maintaining that information, their directory service protocols, or the end-user's client access protocol.</p>

<i>Title</i>	<b>Best Current Practice for the Internet White Pages Service</b>
<i>I-D</i>	<i>draft-ietf-ids-ds-bcp (v5)</i>

<i>Description</i>	<p>This document makes the following recommendations for organizations on the Internet:</p> <ul style="list-style-type: none"> <li>- An organization SHOULD publish public E-mail addresses and other public address information about Internet users within their site.</li> <li>- Most countries have laws concerning publication of information about persons. The currently preferable way for publishing the information is by using X.500 as its data structure and naming scheme. The organization MAY additionally publish it using additional data structures such as whois++.</li> <li>- The organization SHOULD make the published information available to LDAP clients, by allowing LDAP servers access to their data".</li> <li>- The organization SHOULD NOT attempt to charge for simple access to the data.</li> </ul> <p>In addition, it makes the following recommendations for various and sundry other parties:</p> <ul style="list-style-type: none"> <li>- E-mail vendors SHOULD include LDAP lookup functionality into their products, either as built-in functionality or by providing translation facilities</li> <li>- Internet Service providers SHOULD help smaller organizations follow this recommendation, either by providing services for hosting their data, by helping them find other parties to do so, or by helping them bring their own service on-line. All interested parties SHOULD make sure there exists a core X.500 name space in the world, and that all names in this name space are resolvable. (National name spaces may elaborate on the core name space).</li> </ul>
--------------------	---

<i>Title</i>	<b>A Taxonomy of Methods for LDAP Clients Finding Servers</b>
<i>I-D</i>	<i>draft-ietf-ldapext-ldap-taxonomy (v2)</i>
<i>Description</i>	There are several different methods for a LDAP client to find a LDAP server. This draft discusses these methods and provides pointers for interested parties to learn more about implementing a particular method.

<i>Title</i>	<b>Discovering LDAP Services with DNS</b>
<i>I-D</i>	<i>draft-ietf-ldapext-locate (v4)</i>
<i>Description</i>	An LDAP request must be directed to an appropriate server for processing. This document specifies a method for discovering such servers using information in the Domain Name System.

<i>Title</i>	<b>Referrals in LDAP Directories</b>
<i>I-D</i>	<i>draft-ietf-ldapext-refer (v0)</i>

<i>Description</i>	This document defines two reference attributes and associated "referral" object class for representing generic knowledge information in LDAP directories. The attribute uses URIs to represent knowledge, enabling LDAP and non-LDAP services alike to be referenced. The object class can be used to construct entries in an LDAP directory containing references to other directories or services. This document also defines procedures directory servers should follow when supporting these schema elements and when responding to requests for which the directory server does not contain the requested object but may contain some knowledge of the location of the requested object.
--------------------	---

<i>Title</i>	<b>Named References in LDAP Directories</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-namedref (v0)</i>
<i>Description</i>	This document defines schema and protocol elements for representing and manipulating generic knowledge information in LDAP directories. An attribute type "ref" is used to store URIs which may refer to LDAP and non-LDAP services. An object class "referral" is used to construct entries in an LDAP directory which references to other directories or services. A control, ManageDsaIT, is defined to allow clients to manipulate referral objects as normal entries. The document describes procedures directory servers should follow when supporting these elements.

<i>Title</i>	<b>Named References in LDAP Directories</b>
<i>I-D</i>	<i>draft-zeilenga-ldap-namedref (v0)</i>
<i>Description</i>	The OpenLDAP Project is operating an experimental LDAPv3 referral service known as the "OpenLDAP Root Service." The automated system generates referrals based upon service location information published in DNS SRV [RFC2782] resource records. This document describes this service.





---

## Appendix C. Domino Directory forms

This is a summary of the Domino Directory document types, what they are used for, how to configure them, and some real world examples.

---

### C.1 Certificates

A certificate is a unique electronic stamp that identifies a user or a server. In the Notes environment, the server and user IDs contain one or more Notes certificates. In addition, user IDs may contain one or more internet certificates and internet cross certificates that identify users when they use SSL to connect to an internet server or send a signed or encrypted S/MIME mail message.

Certificates are stored in user and server ID files and in the Person, Server, and Certifier documents in the Domino Directory, as well as in the personal address book on the user's workstation.

A certificate contains:

- The name of the certifier that issued the certificate
- The name of the user or server to whom the certificate was issued
- A public key that is stored in both the Domino Directory and the ID file
- An electronic signature
- The expiration date of the certificate

There are four types of certificates that are stored in the Domino Directory: Notes certificates, Notes cross certificates, internet certificates, and internet cross certificates.

#### C.1.1 Notes certificates

When you set up and configure the first server in the domain, Domino creates an organization certifier ID on the server. Domino also adds a certifier document in the Domino Directory. When you register additional organizational units, the new organizational certifier document in the Domino Directory will store the name of the certifier ID that issued it, the signed public key and any certificates held by the certifying ID.

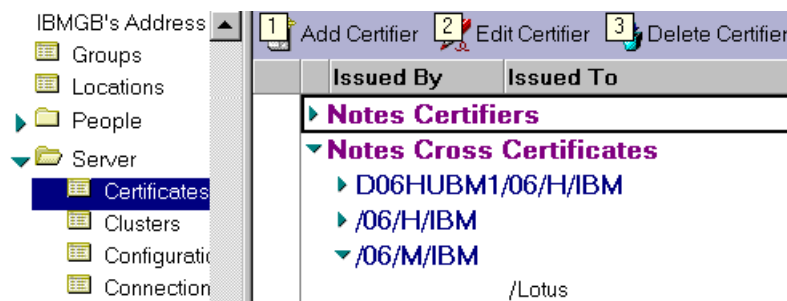
The Domino certificate authority issues certificates that are signed by the organization or organizational unit certifier ID. It creates a public and private key pair while registering the user. The certifier that is used when registering a user determines which certificates will be included in the user ID. In a

diverse organization, organizational unit certifiers are used when you need granularity in name differentiation, security and administration.

### C.1.2 Notes cross certificates

By creating a Notes cross certificate, you allow users in different hierarchical organizations to access servers, to receive signed mail messages and to verify the digital signature of a user from another organization. The Notes server will store the cross certificate in the Domino Directory while a user will store cross certificates in his personal address book. The difference between a cross certificate in the Domino Directory and one in the personal address book is that any cross certificate that exists in the personal address book will only apply to that specific user. Cross certificates in the Domino Directory apply to all users who are part of that hierarchical naming convention.

Cross certification can occur at various levels of an organization. For example, IBM in the United Kingdom has cross certified with Lotus at their mail server level:



This allows Lotus users to access all IBM United Kingdom mail servers (\* /06/M/IBM) to do calendar lookups and so on. At the same time, it also prevents Lotus users to access IBM United Kingdom mail hubs (\* /06/H/IBM) because that level of access might only be needed by administrators in the United Kingdom.

Two way cross certification does not need to be symmetric. For example, in the above case, IBM in the United Kingdom has cross certified with Lotus at their hierarchical organization level (06/M/IBM). Lotus might choose to cross certify with IBM at organizational level (/Lotus with /IBM), which will allow all IBM users access to all Lotus servers, as long as each server is configured to allow \*/IBM access.

### **C.1.3 Internet certificates**

Internet certificates are required when sending encrypted or electronically signed S/MIME mail messages and when using SSL to authenticate a client or server. These certificates validate the identity of a user or server. By using an internet certificate, the recipient of an encrypted S/MIME message knows that the sender's certificate can be trusted and that the certificate used to sign an S/MIME message is valid. It also validates the identity of a server when a Notes client uses SSL to access an internet server.

To issue internet certificates to all users, using the automated processes in Domino, the ideal solution is to get an internet certificate for your organization from a third party certificate authority. This means you will be issuing certificates based on a root certificate that might be more widely known and trusted than your organization.

If you need to issue internet certificates for Notes clients, it can be done by using the existing public and private keys in the Notes ID file and adding it to the user's person document. Using the Domino Directory to issue internet certificates simplifies the process of distributing internet certificates to users.

To be successful when issuing internet certificates, the administration process must be active on the server on which you issue internet certificates and the users must have an internet address specified in their person documents. Notes automatically adds internet certificates stored in the person document to the Notes ID file the next time the user authenticates with the server.

The alternative to this is that each user will have to request their own internet certificates from a third party certificate authority. This is a manual process whereby each user will have to manually copy the public key from the internet certificate into his person document in the Domino Directory.

### **C.1.4 Internet cross certificates**

An internet cross certificate gets issued when one certificate authority requests a certificate from another certificate authority; for example, if you set up the Domino certificate authority and you cross certify your organization with a third party certificate authority. Once the two parties have cross certified, each side trusts the other to issue certificates to users and servers lower in the hierarchical name tree. The copy of the certificate authority's certificate is referred to as a trusted root certificate. To distribute internet cross certificates to all users in your local organization, your local Domino certificate authority would use the above approach.

If you are setting up server authentication for a Notes client, you add this trusted root to a Domino Directory that users can access to generate a cross certificate in their personal address book. To enable a Notes user to automatically pick up internet cross certificates, the certificate must be issued by a common ancestor before Notes will copy the cross certificates to the user's personal address book.

After obtaining the internet cross certificate for the trusted root certificate, the Notes client will trust the certificate authority and by extension, any certificates issued by this certificate authority. This means the Notes client will not get prompted when accessing any servers that have this common root certificate.

For SSL client authentication, the Notes client must have:

- An internet certificate from a Domino or third-party certificate authority
- A trusted root certificate for a Domino or third-party certificate authority to be able to create an internet cross certificate
- A cross certificate for the Domino or third-party certificate authority created from the trusted root certificate

For SSL client authentication, internet clients must have:

- An internet certificate issued from a Domino or third-party certificate authority
- A cross certificate for the Domino or third-party certificate authority created from the trusted root certificate

---

## C.2 Configuration settings

A configuration settings document allows you to set configurations for Notes mail routing via NRPC, mail routing via SMTP, LDAP and the notes.ini file on multiple servers in your domain by using one document. For example, in a hub and spoke topology, you might want to create one group in the Domino Directory where you include all your spoke server names and another group where you include your hub surveillances. Now you would only require two different configuration documents to standardize a whole domain.

Each setting you set applies to every server included in the configuration settings document. Therefore, you need multiple configuration documents if you need different settings for specific servers. To specify additional settings for a specific server that is included in a group, create a separate configuration settings document for that server. For example, if you have an application server that is mail-enabled and the mail-in application is not of any importance, you might restrict the number of mail threads on the server by

using a configuration document. The document that is most specific in terms of which servers it applies to will take precedence.

A server checks the configuration settings documents in the following order:

1. A document specific to the server
2. A group document for any group the server is in
3. The default document

---

### **C.3 Connection documents**

Connection documents provide servers with the information necessary to connect to other servers for mail routing and replication purposes. A connection document provides two types of information: network information and schedule information. The network information defines which server to connect to, how that connection is made and what protocol to use. The schedule information defines when and how often activities—like replication and routing to a particular server—are performed.

---

### **C.4 Domain documents**

Domain documents define domains that are used in mail routing and calendaring and scheduling. Servers that are not in the same Notes named network and in the same domain need domain documents to work out how to route mail outside the home domain.

Depending on each organization's setup, the solution may have a combination of any of the following documents:

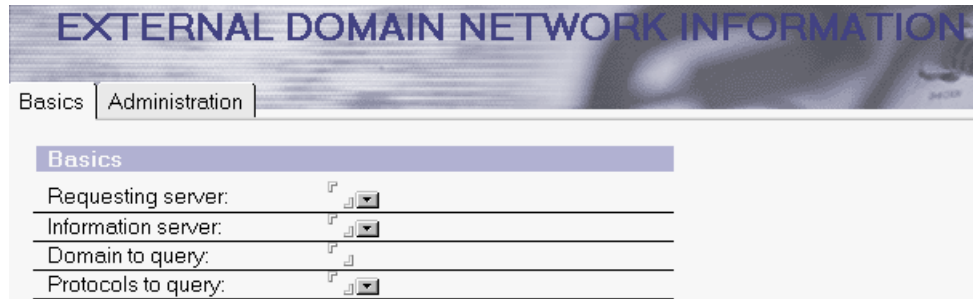
- Foreign domain documents
- Non-adjacent domain documents
- Adjacent domain documents
- Foreign X.400 domain documents
- Foreign SMTP domain documents
- Foreign cc:Mail domain documents
- Global domain documents





---

### **C.5 External domain network information**

External domain network information documents allow users to connect easily to servers outside of their domain. Without these documents, Notes

users will need connection documents to supply network addresses for connections made to servers in an external domain.



EXTERNAL DOMAIN NETWORK INFORMATION	
Basics Administration	
Basics	
Requesting server:	
Information server:	
Domain to query:	
Protocols to query:	

The external domain network information document needs the getadrs server program, which has to be set up as a server add-in program before it can be used.

These documents contains names and addresses of servers from an external domain that use a particular protocol. When users try to connect to a server in an external domain, their home server provides information from these documents to make the connection.

Once these documents have been created, the requesting server in your home domain will contact the information server in the external domain, on set schedules, to update address information. It then places the information in an administration request for processing. When this request is processed, the information is added into a response to the external domain network information document in the Domino Directory.

The external domain server lookup resolves the client's specific request only, it will not return with a list of available servers in the external domain.

---

## C.6 Group documents

Group documents define a list of users and servers for use in access control lists, mail addressing, server access lists and server deny access lists. By using groups you can simplify Domino Directory administration.

For example, in a big organization, for people leaving the company, you can create a group called Terminations with a group type of deny list only. As group members you can use nested groups with names like Terminations 1999 and Terminations 2000 to prevent you from going over the group document level of 15KB. You then need to create these groups, also as deny list only groups. This will add the groups into the "Server\Deny access

groups” view. Administration and referencing these groups then becomes a lot more manageable. The last step needed in this scenario is to add the main Terminations group name in the Not access server field of each Server document.

---

### **C.7 Holiday documents**

Holiday documents provide a way for an organization to have a centrally managed collection of documents that contain information on scheduled holidays or events. Users select the type of holiday documents to import and this adds the information to their personal calendars.

---

### **C.8 Location documents**

These documents contain communication and other location-specific settings for use from a client. It is useful for administrators who also use the Domino Directory as their personal address book. These documents define information for various work locations.

The form layout depends on the type of location document you are setting up. For example, if you are setting up a location type of ‘No connection’, you will be presented with five sections to complete: basics, mail, internet browser, advanced and administration. In this case you will see a replication section, but you will be unable to configure anything in it. If you are setting up a location type of ‘Notes direct dialup’, you will be presented with nine sections to complete: basics, servers, ports, mail, internet browser, replication, phone settings, advanced and administration.

---

### **C.9 Mail-in database documents**

If an application is designed to receive mail, you will need to create a mail-in database document in the Domino Directory, which will define the location of the database.

A typical example of a mail-in database is when you configure a Domino server to report statistics and you specify a mail address. In a big

organization, an administrator will be flooded with statistics, so a mail-in database is an easy way around this problem. Here is an example:

**MAIL-IN DATABASE: D06ML017 Stats**

Basics | Database Information | Other | Administration

**Basics**

Mail-in name:	D06ML017 Stats/IBM
Internet message storage:	No Preference
Internet Address:	
Description:	The Stats Mail-In DB for server D06ML017/06/M/IBM.

**MAIL-IN DATABASE: D06ML017**

Basics | Database Information | Other | Administration

**Location**

Domain:	IBMGB
Server:	D06ML017/06/M/IBM
Filename:	stats717.nsf

In this example, the server's mail-in name will be "D06ML017/06/M/IBM Stats". The administrator specifies a mail address of "D06ML017@IBMGB" or "D06ML017/06/M/IBM Stats@IBMGB" and the mail router will deliver the statistics reports to the mail-in database called stats717.nsf.

If multiple replicas of a mail-in database exist, a mail-in document is needed for every server that has a replica of the database.

---

## C.10 Person documents

A person document describes a Notes or non-Notes user in the Domino Directory. This is the default form. At user registration time, a person document is created by the registration process. Notes uses this document to identify and locate each user.

This form consists of different sections, which include the user's first name, middle initial, last name, distinguished name, short nickname and any other aliases as well as business details and public key information.



---

## C.11 Program documents

Program documents determine when and on which server Domino server tasks and other programs will run.

As an example, on application servers in IBM, most Notes housekeeping tasks are set up to run via program documents. This is done to maximize server availability and minimize user interruptions. Here are some examples.

The compact program on a specific subdirectory can be set up like this:

The screenshot shows the 'PROGRAM: compact' configuration window with the 'Basics' tab selected. The window has a title bar and three tabs: 'Basics', 'Schedule', and 'Administration'. The 'Basics' tab is active, displaying a table with the following information:

Program:	compact
Parameters:	a_dir
Server to run on:	D06DBL28/06/A/IBM
Comments:	

The screenshot shows the 'PROGRAM: compact' configuration window with the 'Schedule' tab selected. The window has a title bar and three tabs: 'Basics', 'Schedule', and 'Administration'. The 'Schedule' tab is active, displaying a table with the following information:

Enabled/disabled:	ENABLED
Run at times:	19:00 each day
Repeat interval of:	0 minutes
Days of week:	Tue

The designer server task can be removed from the ServerTasksAt1 line in the notes.ini file, to limit user impact, and can then be set up as a program document to only run when scheduled. By default a program document is

enabled, but you can also create it as disabled and then enable the document when needed.

The screenshot shows the 'PROGRAM: Design' form with the 'Basics' tab selected. The form has a header bar with the title 'PROGRAM: Design' and three tabs: 'Basics', 'Schedule', and 'Administration'. Below the tabs, the 'Basics' section contains four fields: 'Program:' with a dropdown menu showing 'Design', 'Parameters:' with a dropdown menu, 'Server to run on:' with a dropdown menu showing 'D06DBL28/06/A/IBM', and 'Comments:' with a text area.

The screenshot shows the 'PROGRAM: Design' form with the 'Schedule' tab selected. The form has a header bar with the title 'PROGRAM: Design' and three tabs: 'Basics', 'Schedule', and 'Administration'. Below the tabs, the 'Schedule' section contains four fields: 'Enabled/disabled:' with a dropdown menu showing 'DISABLED', 'Run at times:' with a dropdown menu showing '03:00 each day', 'Repeat interval of:' with a dropdown menu showing '0 minutes', and 'Days of week:' with a dropdown menu showing 'Mon'.

---

## C.12 Resource documents

The resource documents define resources that Notes clients can reserve by using the calendar and scheduling features. After you create a resource document, the information that you can change includes the availability settings, description and ownership options fields.

New resource information is not available until the administration process updates the resource document and the change has replicated to all servers in the domain that schedule resources.

---

## C.13 Server documents

The server document form is the heart of the Domino server configuration for network configuration, security, mail routing configuration settings,

transaction logging settings, and internet settings. As such, it is discussed at length in the on-line Domino documentation “Domino 5 Administration Help” as well as in the default Domino manuals “Administering the Domino System” Volumes I and II.

---

## **C.14 User setup profile documents**

The user profile documents define a standard set of configuration options for Notes clients at setup time, or at the next time the user connects after you made changes to these documents.

The type of information that you can push to users is settings like the default pass-through server, internet server and configurations, and specific databases that will appear on each user's workspace, either as links or that will be created as local replicas.

As organizations start expanding their Domino Directory services, these documents provide the ideal opportunity to deploy a user Directory Catalog painlessly throughout the organization.



## Appendix D. Syntax of LDAPSearch command

Here are the syntax options for the LDAPSearch command:

Table 6. LDAPSearch parameters

Parameter	Use to
-?	Print help on using ldapsearch.
-a <i>deref</i>	Specify alias de-referencing. Enter never, always, search, or find. Never is the default if you don't use this parameter.
-A	Retrieve only attribute names, not the values for the attributes.
-b <i>base dn</i>	Specify a distinguished name to use as the starting point for beginning the search. Use quotation marks to specify the value - for example, "ou=West,o=Acme,c=US". You must use this parameter if the server you're searching requires you to specify a search base. Otherwise, it is optional. Optionally use -s along with -b to determine the scope of the search. Without -s, -b searches the entry specified as the starting point and all descendants of the entry.
-D <i>bind dn</i>	Specify a distinguished name that the server uses to authenticate you. The name must correspond to an entry in the directory and must have the necessary access to search the directory. Specify the name in quotation marks -- for example, "cn=Directory Manager,o=Acme,c=US". If you don't use this parameter, the connection to the server occurs anonymously. You must use -D if the server doesn't allow anonymous connections. Along with -D, you must use the -w parameter to specify a password associated with the distinguished name.
-f <i>file</i>	Specify a file that contains search filters to use -- for example, -f filters.
-h <i>host name</i>	Specify the host name of the server to which you're connecting -- for example, -h server.acme.com. This parameter is required.
-l <i>timelimit</i>	Specify a time limit (in seconds) for the search to complete. If you do not specify this parameter or if you specify a limit of 0, searches can take an unlimited amount of time. ldapsearch never waits longer than the search time limit set on the server, however.
-L	Specify that the output is in LDIF format. LDIF format uses a colon (:) as the attribute delineator rather than an equal sign (=). Using LDIF format lets you, for example, import the contents of the output into an LDAP-compliant directory.

Parameter	Use to
-n	Specify that ldapsearch not actually search but instead only show how it would perform the search.
-p <i>port</i>	Specify the port that the server uses. If you don't use this parameter, ldapsearch uses port 389 by default.
-s <i>scope</i>	Specify the scope of the search when you use the -b parameter:  base -- to search only the entry specified with the -b parameter onelevel -- to search only the immediate children of the entry specified with the -b parameter but not the entry itself subtree -- to search the entry specified with the -b parameter and all of its descendants. This is the default behavior when you use -b without -s.  The order in which you specify -b and -s is unimportant.
-S <i>attribute</i>	Sort the results by a specified attribute.
-z <i>sizelimit</i>	Specify the maximum number of entries to return. If you don't specify this parameter or if you specify a limit of 0, an unlimited number of entries are returned. ldapsearch never returns more entries than the server allows, however.
-u	Specify that ldapsearch return distinguished names in a user-friendly format.
-v	Specify that ldapsearch run in verbose mode.
-w <i>password</i>	Specify the password associated with a distinguished name used with the -D parameter.

---

## Appendix E. Description of test environment

The environment that we have set up for testing consists of the following servers and services:

---

### E.1 balder.lotus.com

IBM Netfinity 3000 Server

Microsoft Windows NT 4, Service Pack 4

Lotus Domino Release 5.0.4

HTTP server listening on port 8080

LDAP server listening on port 3890

iPlanet Directory Server Version 4.1.2

LDAP server listening on port 389

iPlanet Web Server Version 4.1

HTTP server listening on port 80

Exchange Server Version 5.5

LDAP server listening on port 3899

Microsoft Internet Information Server Version 2.0

HTTP server listening on port 8000

---

### E.2 odin.lotus.com

IBM Personal Computer 300PL

Microsoft Windows NT 4, Service Pack 6

Lotus Domino Release 5.0.4

Novell eDirectory Version 8 for NT

LDAP server listening on port 389

---

### **E.3 heimdal.lotus.com**

IBM Netfinity 3000 Server

Microsoft Windows 2000 Advanced Server

Active Directory

LDAP server listening on port 389

Internet Information Server

HTTP server listening on port 80

Lotus Domino Release 5.0.4

LDAP server listening on port 3890

HTTP server listening on port 82

Microsoft Exchange 2000 Release Candidate 1

---

### **E.4 gefion.lotus.com**

IBM Personal Computer 300PL

Microsoft Windows NT 4.0, Service Pack 6

Lotus Domino Release 5.0.4

HTTP server listening on port 80

IBM SecureWay Directory 3.1.1 for NT

LDAP server listening on port 389

IBM WebSphere Application Server V3



---

## Appendix F. Sample code using the Notes API

The Notes API is a C language programming interface for Lotus Domino. The API can be used to access all of the different elements of Notes, including the Domino Directory. You can find the latest version of the Notes API at:

<http://www.lotus.com/developer>

With the Notes API, you have low-level access to the internals of any Domino database. The Domino Directory is no exception. You can use the API to add both documents and design elements, and in so doing add entries to the directory, or extend the directory schema.

Here's an example of Notes API code to add a subform to the directory and add a field to the subform, thereby extending the domino schema. It uses version 5.0.4 of the Notes API. This code performs the same operation that we did manually in 4.2.4.1, "Adding new attributes to the existing schema" on page 53. Using the Notes API to generate design elements is quite complicated. We recommend that you use the Domino Designer UI whenever possible.

```
/* A program to update the schema of the */
/* Domino Directory with the Notes API   */

/* C Headers */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/types.h>

/*Notes API Headers */

#include <lapicinc.h>
#include <lapiplat.h>
#include <global.h>
#include <nsfdata.h>
#include <osfile.h>
#include <nsfdb.h>
#include <nsfnote.h>
#include <stdnames.h>
#include <osmem.h>
#include <editods.h>
#include <colorid.h>
#include <editdflt.h>
```

```

#include <fontid.h>
#include <nsfsearc.h>
#include <easycd.h>

#define CD_BUFFER_LENGTH      64000

/*Main Program */

LAPI_MAIN
{

    /* variable declarations */

    char fullpath[MAXPATH];
    DBHANDLE hDB;
    char Subformname[]="Acme Person";
    char PersonFormName[]="$PersonExtensibleSchema";
    STATUS result=NOERROR;
    NOTEHANDLE hSubFormNote;
    WORD InfoBufferLength;
    HANDLE hBuffer;
    char far *pBufferStart;
    char far *pBuffer;
    BYTE InfoLengthBytes;
    CDDOCUMENT InfoCDStruct;
    DWORD ItemLength;
    char AcmePersonFieldLabel[]="Acme Person";
    char AcmePersonFormula[] =
        "FIELD $objectclass:=$objectclass:\"AcmePerson\";1";
    HANDLE hAcmePersonFormulaComp;
    char *pAcmePersonFormulaComp;
    WORD AcmePersonFormulaCompLength;
    char AcmePersonFieldName[]="AcmePerson";
    char AcmePersonFieldDesc[]="Used to extend schema";
    char EyeColorFieldLabel[]="Eye Color";
    char EyeColorFieldName[]="EyeColor";
    char EyeColorFieldDesc[]="Enter this person\'s eye color.";
    char far FlagData[5];
    WORD FlagLength;
    WORD CDBufferLength=MAXONESEGSIZE;
    CDPABDEFINITION ParagraphStyleDef;
    CDPARAGRAPH Paragraph;
    CDPABREFERENCE ParagraphStyle;
    FONTIDFIELDS *pFontID;
    CDTEXT FieldLabels;
    CDBEGINRECORD CDBegin;
    CDEXT2FIELD CDExt2Field;

```

```

CDFIELD CDField;
CDENDRECORD CDEnd;
STATUS CompileError;
WORD CompileErrorLine;
WORD CompileErrorColumn;
WORD CompileErrorOffset;
WORD CompileErrorLength;
WORD ClassForm = NOTE_CLASS_FORM

/* Start the Notes system */

LAPI_INIT(result);

/* Build path to directory template */

result=OSPathNetConstruct
    (NULL,"balder/RedBook","pubnames.ntf",fullpath);

/* Open DomDir template */

printf("opening domino directory");
result=NSFDbOpen(fullpath,&hDB);

/* Create subform note */

printf("creating note for subform");
result=NSFNoteCreate(hDB,&hSubFormNote);

/* Create fields on subform note */

/* Create $Title item - the name of the subform*/

printf("creating $Title");
result=NSFItemSetText
    (hSubFormNote,ITEM_NAME_TEMPLATE_NAME,Subformname,MAXWORD);

/* Create $Info item */

/* Create a buffer for the CDDocument */

InfoBufferLength = ODSLength(_CDDOCUMENT);
result=OSMemAlloc(0,InfoBufferLength,&hBuffer);

/* Set a pointer to the start of the buffer */

pBufferStart=OSLockObject(hBuffer);
memset (pBufferStart,0,(size_t) InfoBufferLength);

```

```

pBuffer=pBufferStart;
InfoLengthBytes= (BYTE) ODSLength(_CDDOCUMENT);

/* Build the CDDOCUMENT structure */

memset(&InfoCDStruct,0,sizeof(CDDOCUMENT));
InfoCDStruct.Header.Signature=SIG_CD_DOCUMENT;
InfoCDStruct.Header.Length=InfoLengthBytes;
InfoCDStruct.PaperColor=NOTES_COLOR_WHITE;
InfoCDStruct.FormFlags=0;
InfoCDStruct.NotePrivileges=0;
InfoCDStruct.FormFlags2=0;
InfoCDStruct.InherFieldNameLength=0;
InfoCDStruct.PaperColorExt=0;

/* Write CDDOCUMENT to buffer */

ODSWriteMemory ( (void far * far *)&pBuffer,
_CDDOCUMENT,&InfoCDStruct,1);

/* Store buffer in $Info item*/

ItemLength = (DWORD) InfoBufferLength;
result=NSFItemAppend(hSubFormNote,
    0,
    ITEM_NAME_DOCUMENT,
    (WORD) strlen(ITEM_NAME_DOCUMENT),
    TYPE_COMPOSITE,
    (void far *)pBufferStart,
    ItemLength);

/*Clean Up */

OSUnlockObject(hBuffer);
OSMemFree(hBuffer);

/* Create $Flags item - to designate it a subform */

FlagData[0]=DESIGN_FLAG_ADD;
FlagData[1]=DESIGN_FLAG_NO_COMPOSE;
FlagData[2]=DESIGN_FLAG_SUBFORM;
FlagData[3]=DESIGN_FLAG_HIDE_FROM_V3;
FlagLength=4;
result=NSFItemAppend(hSubFormNote,
    ITEM_SUMMARY,
    DESIGN_FLAGS,
    (WORD) strlen(DESIGN_FLAGS),

```

```

        TYPE_TEXT,
        (void*)&FlagData[0],
        FlagLength);

/* Create placeholder items - used for view column creation*/

result=NSFItemAppend(hSubFormNote,
    ITEM_PLACEHOLDER,
    AcmePersonFieldName,
    (WORD) strlen(AcmePersonFieldName),
    TYPE_INVALID_OR_UNKNOWN,
    NULL,0);
result=NSFItemAppend(hSubFormNote,
    ITEM_PLACEHOLDER,
    EyeColorFieldName,
    (WORD) strlen(EyeColorFieldName),
    TYPE_INVALID_OR_UNKNOWN,
    NULL,0);

/* Create $Body item - contains the structure */
/* of the form and the fields on it          */

/* Create buffer for $Body field */

result=OSMemAlloc(0,CDBufferLength, &hBuffer);
pBufferStart=(char far *)OSLockObject(hBuffer);
memset(pBufferStart,0,(size_t) CDBufferLength);
pBuffer=pBufferStart;

/* Set up Paragraph definition */

memset(&ParagraphStyleDef,0,sizeof(_CDPABDEFINITION));
ParagraphStyleDef.Header.Signature=SIG_CD_PABDEFINITION;
ParagraphStyleDef.Header.Length=
    ODSLength(_CDPABDEFINITION);
ParagraphStyleDef.PABID=1;
ParagraphStyleDef.JustifyMode=JUSTIFY_LEFT;
ParagraphStyleDef.LineSpacing=DEFAULT_LINE_SPACING;
ParagraphStyleDef.ParagraphSpacingBefore=
    DEFAULT_ABOVE_PAR_SPACING;
ParagraphStyleDef.ParagraphSpacingAfter=
    DEFAULT_BELOW_PAR_SPACING;
ParagraphStyleDef.LeftMargin=DEFAULT_LEFT_MARGIN;
ParagraphStyleDef.RightMargin=DEFAULT_RIGHT_MARGIN;
ParagraphStyleDef.FirstLineLeftMargin=
    DEFAULT_FIRST_LEFT_MARGIN;
ParagraphStyleDef.Tabs=DEFAULT_TABS;

```

```

ParagraphStyleDef.Tab[0]=DEFAULT_TAB_INTERVAL;
ParagraphStyleDef.Flags=0;
ParagraphStyleDef.TabTypes=TAB_DEFAULT;
ParagraphStyleDef.Flags2=DEFAULT_FLAGS2;

/* Write paragraph definition to buffer */

ODSWriteMemory ((void far * far *)&pBuffer,
    _CDPABDEFINITION,&ParagraphStyleDef,1);

/* Start new paragraph */

Paragraph.Header.Signature=SIG_CD_PARAGRAPH;
Paragraph.Header.Length=(BYTE) ODSLength(_CDPARAGRAPH);
ODSWriteMemory ((void far * far *)&pBuffer,
    _CDPARAGRAPH,&Paragraph,1);

/* Tell paragraph to use definition above */

ParagraphStyle.Header.Signature=
    (BYTE) SIG_CD_PABREFERENCE;
ParagraphStyle.Header.Length=
    (BYTE) ODSLength(_CDPABREFERENCE);
ParagraphStyle.PABID=1;
ODSWriteMemory ((void far * far *)&pBuffer,
    _CDPABREFERENCE,&ParagraphStyle,1);

/* Add AcmePerson field */

/* Insert text for label */

FieldLabels.Header.Signature=SIG_CD_TEXT;
FieldLabels.Header.Length=ODSLength(_CDTEXT) +
    (WORD) strlen(AcmePersonFieldLabel);
pFontID=(FONTIDFIELDS *) &FieldLabels.FontID;
pFontID->Face=FONT_FACE_SWISS;
pFontID->Attrib=ISBOLD;
pFontID->Color=NOTES_COLOR_BLUE;
pFontID->PointSize=9;
ODSWriteMemory((void far * far *)&pBuffer,
    _CDTEXT,&FieldLabels,1);
memcpy( (char *)pBuffer,AcmePersonFieldLabel,
    (WORD) strlen(AcmePersonFieldLabel));
pBuffer += (WORD) strlen(AcmePersonFieldLabel);
if ((pBuffer-pBufferStart) % 2) pBuffer++;

/* Insert field itself */

```

```

CDBegin.Header.Signature=SIG_CD_BEGIN;
CDBegin.Header.Length=(BYTE) ODSLength(_CDBEGINRECORD);
CDBegin.Version=0;
CDBegin.Signature=SIG_CD_FIELD;
ODSWriteMemory((void far * far *)&pBuffer,
    _CDBEGINRECORD, (void far *) &CDBegin,1);
memset(&CDExt2Field,0,sizeof(CDEXT2FIELD));
CDExt2Field.Header.Signature=SIG_CD_EXT2_FIELD;
CDExt2Field.Header.Length= (WORD) ODSLength(_CDEXT2FIELD);
ODSWriteMemory((void far * far *)&pBuffer,
    _CDEXT2FIELD, (void far *) &CDExt2Field,1);
CDField.Header.Signature=SIG_CD_FIELD;
CDField.Flags=0;
CDField.DataType=TYPE_TEXT;
CDField.ListDelim=LDDELIM_SEMICOLON;
CDField.NumberFormat.Digits=0;
CDField.NumberFormat.Format=0;
CDField.NumberFormat.Attributes=0;
CDField.NumberFormat.Unused=0;
CDField.TimeFormat.Date=0;
CDField.TimeFormat.Time=0;
CDField.TimeFormat.Zone=0;
CDField.TimeFormat.Structure=0;
pFontID=(FONTIDFIELDS *)&CDField.FontID;
pFontID->Face=FONT_FACE_SWISS;
pFontID->Attrib=0;
pFontID->Color=NOTES_COLOR_BLACK;
pFontID->PointSize=9;
result=NSFFormulaCompile(NULL,0,AcmePersonFormula,
    (WORD) strlen(AcmePersonFormula),
    &hAcmePersonFormulaComp,
    &AcmePersonFormulaCompLength,
    &CompileError,
    &CompileErrorLine,
    &CompileErrorColumn,
    &CompileErrorOffset,
    &CompileErrorLength);
CDField.DVLength=AcmePersonFormulaCompLength;
CDField.ITLength=0;
CDField.TabOrder=0;
CDField.IVLength=0;
CDField.NameLength=strlen(AcmePersonFieldName);
CDField.DescLength=strlen(AcmePersonFieldDesc);
CDField.TextValueLength=0;
CDField.Header.Length= ODSLength(_CDFIELD)+
    CDField.DVLength+

```

```

        CDField.ITLength+
        CDField.IVLength+
        CDField.NameLength+
        CDField.DescLength+
        CDField.TextValueLength;
if (CDField.Header.Length % 2) CDField.Header.Length++;
ODSWriteMemory ((void far * far *)&pBuffer,
    _CDFIELD, (void far *)&CDField, 1);
pAcmePersonFormulaComp=OSLock(char, hAcmePersonFormulaComp);
memcpy(pBuffer, pAcmePersonFormulaComp, AcmePersonFormulaCompLength);
pBuffer += AcmePersonFormulaCompLength;
OSUnlockObject(hAcmePersonFormulaComp);
OSMemFree(hAcmePersonFormulaComp);
memcpy(pBuffer, AcmePersonFieldName, CDField.NameLength);
pBuffer += CDField.NameLength;
memcpy(pBuffer, AcmePersonFieldDesc, CDField.DescLength);
pBuffer += CDField.DescLength;
if ((pBuffer-pBufferStart) % 2) pBuffer++;
CDEnd.Header.Length = (BYTE)ODSLength(_CDENDRECORD);
CDEnd.Header.Signature = SIG_CD_END;
CDEnd.Version=0;
CDEnd.Signature=SIG_CD_FIELD;
ODSWriteMemory((void far * far *)&pBuffer,
    _CDENDRECORD, (void far *)&CDEnd, 1);

/* New paragraph */

Paragraph.Header.Signature=SIG_CD_PARAGRAPH;
Paragraph.Header.Length=(BYTE) ODSLength(_CDPARAGRAPH);
ODSWriteMemory ((void far * far *)&pBuffer,
    _CDPARAGRAPH, &Paragraph, 1);
ParagraphStyle.Header.Signature=
    (BYTE) SIG_CD_PABREFERENCE;
ParagraphStyle.Header.Length=
    (BYTE) ODSLength(_CDPABREFERENCE);
ParagraphStyle.PABID=1;
ODSWriteMemory ((void far * far *)&pBuffer,
    _CDPABREFERENCE, &ParagraphStyle, 1);

/* Text for field label */

FieldLabels.Header.Signature=SIG_CD_TEXT;
FieldLabels.Header.Length=ODSLength(_CDTEXT) +
    (WORD) strlen(EyeColorFieldLabel);
pFontID=(FONTIDFIELDS *) &FieldLabels.FontID;
pFontID->Face=FONT_FACE_SWISS;
pFontID->Attrib=ISBOLD;

```



```

pFontID->Color=NOTES_COLOR_BLUE;
pFontID->PointSize=9;
ODSWriteMemory((void far * far *)&pBuffer,
    _CDTEXT,&FieldLabels,1);
memcpy( (char *)pBuffer, EyeColorFieldLabel,
    (WORD) strlen(EyeColorFieldLabel));
pBuffer += (WORD) strlen(EyeColorFieldLabel);
if ((pBuffer-pBufferStart) % 2) pBuffer++;

/* Eye Color Field */

CDBegin.Header.Signature=SIG_CD_BEGIN;
CDBegin.Header.Length=(BYTE) ODSLength(_CDBEGINRECORD);
CDBegin.Version=0;
CDBegin.Signature=SIG_CD_FIELD;
ODSWriteMemory((void far * far *)&pBuffer,
    _CDBEGINRECORD, (void far *) &CDBegin,1);
memset(&CDExt2Field,0,sizeof(CDEXT2FIELD));
CDExt2Field.Header.Signature=SIG_CD_EXT2_FIELD;
CDExt2Field.Header.Length= (WORD) ODSLength(_CDEXT2FIELD);
ODSWriteMemory((void far * far *)&pBuffer,
    _CDEXT2FIELD, (void far *) &CDExt2Field,1);
CDField.Header.Signature=SIG_CD_FIELD;
CDField.Flags=FEDITABLE;
CDField.DataType=TYPE_TEXT;
CDField.ListDelim=LDDELIM_SEMICOLON;
CDField.NumberFormat.Digits=0;
CDField.NumberFormat.Format=0;
CDField.NumberFormat.Attributes=0;
CDField.NumberFormat.Unused=0;
CDField.TimeFormat.Date=0;
CDField.TimeFormat.Time=0;
CDField.TimeFormat.Zone=0;
CDField.TimeFormat.Structure=0;
pFontID=(FONTIDFIELDS *)&CDField.FontID;
pFontID->Face=FONT_FACE_SWISS;
pFontID->Attrib=0;
pFontID->Color=NOTES_COLOR_BLACK;
pFontID->PointSize=9;
CDField.DVLength=0;
CDField.ITLength=0;
CDField.TabOrder=0;
CDField.IVLength=0;
CDField.NameLength=strlen(EyeColorFieldName);
CDField.DescLength=strlen(EyeColorFieldDesc);
CDField.TextValueLength=0;
CDField.Header.Length= ODSLength(_CDFIELD)+

```

```

        CDField.DVLength+
        CDField.ITLength+
        CDField.IVLength+
        CDField.NameLength+
        CDField.DescLength+
        CDField.TextValueLength;
    if (CDField.Header.Length % 2) CDField.Header.Length++;
    ODSWriteMemory ((void far * far *)&pBuffer,
        _CDFIELD, (void far *)&CDField, 1);
    memcpy(pBuffer, EyeColorFieldName, CDField.NameLength);
    pBuffer += CDField.NameLength;
    memcpy(pBuffer, EyeColorFieldDesc, CDField.DescLength);
    pBuffer += CDField.DescLength;
    if ((pBuffer-pBufferStart) % 2) pBuffer++;
    CDEnd.Header.Length = (BYTE)ODSLength(_CDENDRECORD);
    CDEnd.Header.Signature = SIG_CD_END;
    CDEnd.Version=0;
    CDEnd.Signature=SIG_CD_FIELD;
    ODSWriteMemory((void far * far *)&pBuffer,
        _CDENDRECORD, (void far *)&CDEnd, 1);

    /* Store buffer in item */

    result=NSFItemAppend(hSubFormNote, 0,
        ITEM_NAME_TEMPLATE,
        (WORD) strlen(ITEM_NAME_TEMPLATE),
        TYPE_COMPOSITE, (void *)pBufferStart,
        pBuffer-pBufferStart);

    /* Create $Fields item */

    result=NSFItemCreateTextList(hSubFormNote,
        "$FIELDS",
        AcmePersonFieldName,
        (WORD) strlen(AcmePersonFieldName));
    result=NSFItemAppendTextList(hSubFormNote,
        "$FIELDS",
        EyeColorFieldName,
        (WORD) strlen(EyeColorFieldName), TRUE);

    /* Set NoteClass to Form */

    NSFNoteSetInfo(hSubFormNote, _NOTE_CLASS, &ClassForm);
    result=NSFNoteUpdate(hSubFormNote, 0);

    /* Clean up */

```

```

OSUnlockObject(hBuffer);
OSMemFree(hBuffer);
NSFNoteClose(hSubFormNote);

/* Insert subform note into $PersonExtensibleSchema form */

result=SubformInsert(hDB,Subformname,PersonFormName,0);
result=NSFDbClose(hDB);

/* End program */

LAPI_RETURN(result);
}

```



---

## Appendix G. Basic directory concepts

People and businesses are increasingly relying on networked computer systems to support distributed applications. These distributed applications might interact with computers on the same local area network (LAN), within a corporate intranet, or anywhere in the world on the Internet. To improve functionality, provide ease of use, and enable cost-effective administration of distributed applications, information about the services, resources, users, and other objects accessible from the applications needs to be organized in a clear and consistent manner. Much of this information can be shared among many applications, but it must also be protected to prevent unauthorized modification or the disclosure of private information.

Information describing the various users, applications, files, printers, and other resources accessible from a network is often collected into a special database, called a directory. As the number of different networks and applications has grown, the number of specialized directories of information has also grown, resulting in islands of information that cannot be shared and are difficult to maintain. If all of this information could be maintained and accessed in a consistent and controlled manner, it would provide a focal point for integrating a distributed environment into a consistent and seamless system.

This chapter defines standards and terminologies involved in accessing Directories. This provides a framework for discussing Domino Directory definitions and the part that a Domino Directory can play in an integrated environment.

---

### G.1 What is a directory?

A directory is a listing of information about objects arranged in some order and that gives details about each object. Common examples are a city telephone directory and a library card catalog. For a telephone directory, the objects listed are people; the names are arranged alphabetically, and the details given about each person are address and telephone number. Books in a library card catalog are ordered by author or by title, and information such as the ISBN attribute of the book and other publication information is given.

In computer terms, a directory is a specialized database, also called a data repository, that stores typed and ordered information about objects. A particular directory might list information about printers (the objects) consisting of typed information such as location (a formatted character

string), speed in pages per minute (numeric), print streams supported (for example PostScript or ASCII), and so on.

Directories allow users or applications to find resources that have the characteristics needed for a particular task. For example, a directory of users can be used to look up a person's e-mail address or fax number. A directory could be searched to find a nearby PostScript color printer. Or a directory of application servers could be searched to find a server that can access customer billing information.

The terms *white pages* and *yellow pages* are particular directory applications. If the name of an object (person, printer) is known, its characteristics (phone number, pages per minute) can be retrieved. This is similar to looking up a name in the white pages of a telephone directory. On the other hand, if the name of a particular individual attribute is not known, the directory can be searched for a list of objects that meet a certain requirement. This is like looking up a listing of hairdressers in the yellow pages of a telephone directory. However, directories stored on a computer are much more flexible than the yellow pages of a telephone directory, because they can usually be searched by a range of criteria, not just by a single predefined set of categories.

### **G.1.1 Directory clients and servers**

Directories are usually accessed using the client/server model of communication. An application that wants to read or write information in a directory does not access the directory directly. Instead, it calls a function or application programming interface (API) that causes a message to be sent to another process. This second process accesses the information in the directory on behalf of the requesting application. The results of the read or write are then returned to the requesting application (see Figure 70 on page 219).

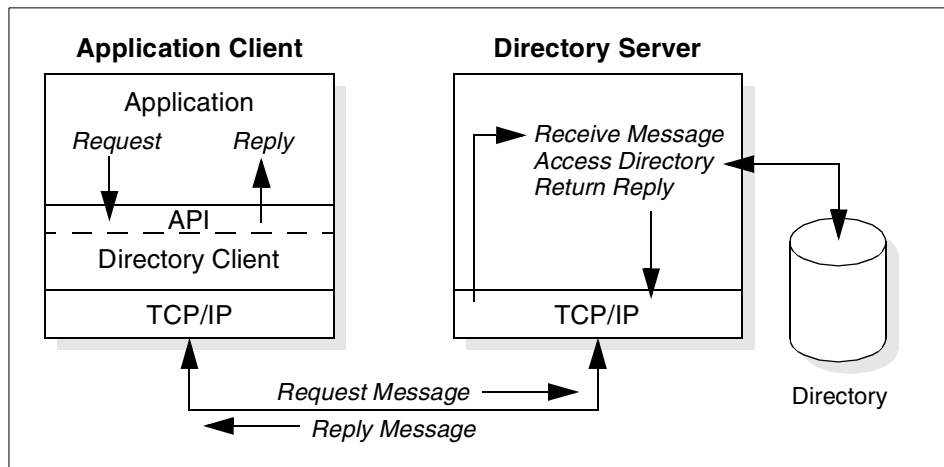


Figure 70. Directory client/server interaction

The request is performed by the directory client, and the process that maintains and looks up information in the directory is called the directory server. In general, servers provide a specific service to clients. Sometimes, a server might become the client of other servers in order to gather the information necessary to process a request.

A directory service is only one type of service that might be available in a client/server environment. Other common examples of services are file services, mail services, print services, Web page services, and so on. The client and server processes might or might not be on the same machine. A server is capable of serving many clients. Some servers can process client requests in parallel. Other servers queue incoming client requests for serial processing if they are currently busy processing another client's request.

An API defines the programming interface that a particular programming language uses to access a service. The format and contents of the messages exchanged between client and server must adhere to an agreed-upon protocol. LDAP defines a message protocol used by directory clients and directory servers. There are also associated LDAP and JDAP APIs for C language, and ways to access the directory from a Java application using JNDI (see 6.3, "Using application development tools with Domino Directory services" on page 105). The client is not dependent upon a particular implementation of the server, and the server can implement the directory however it chooses.

## G.1.2 Distributed Directories

The terms *local*, *global*, *centralized*, and *distributed* are often used to describe a directory or directory service. These terms mean different things to different people in different contexts. In this section, these terms are explained as they apply to directories in different contexts.

In general, *local* means something is close by, and *global* means that something is spread across the universe of interest. The universe of interest might be a company, a country, or the Earth. Local and global are two ends of a continuum. That is, something may be more or less global or local than something else. *Centralized* means that something is in one place, and *distributed* means that something is in more than one place. Like local and global, something can be distributed to a greater or lesser extent.

The information stored in a directory can be simultaneously local and global in scope. For example, a directory that stores local information might consist of the names, e-mail addresses, public encryption keys, and so on of members of a department or workgroup. A directory that stores global information might store information for an entire company. Here, the universe of interest is the company.

The clients that access information in the directory can be local or remote. Local clients may all be located in the same building or on the same LAN. Remote clients might be distributed across the continent or planet.

The directory itself can be *centralized* or *distributed*. If a directory is centralized, there may be one directory server at one location or a directory server that hosts data from distributed systems. If the directory is distributed, there are multiple servers, usually geographically dispersed, that provide access to the directory.

When a directory is distributed, the information stored in the directory can be *partitioned* or *replicated*. When information is partitioned, each directory server stores a unique and non-overlapping subset of the information. That is, each directory entry is stored by one and only one server. One of the techniques to partition the directory is to use LDAP referrals. LDAP referrals allow the users to refer LDAP requests to either the same or different name spaces stored in a different (or the same) server. When information is replicated, the same directory entry is stored by more than one server. In a distributed directory, some information may be partitioned while some may be replicated.

The three *dimensions* of a directory—scope of information, location of clients, and distribution of servers—are independent of each other. For example,



clients scattered across the globe can access a directory containing only information about a single department, and that directory can be replicated at many directory servers. Or, clients in a single location can access a directory containing information about everybody in the world that is stored by a single directory server.

The scope of information to be stored in a directory is often given as an application requirement. The distribution of directory servers and the way in which data is partitioned or replicated can often be controlled to effect the performance and availability of the directory. For example, a distributed and replicated directory might perform better because a read request can be serviced by a nearby server. A centralized directory may be less available because it is a single point of failure. However, a distributed directory might be more difficult to maintain because multiple sites, possibly under the control of multiple administrators, must be kept up-to-date and in running order.

The design and maintenance of a directory service can be complex, and many trade-offs may be involved. You can see more detailed information about the topic in the following chapters.

### **G.1.3 Directory security**

The security of information stored in a directory is a major consideration. Some directories are meant to be accessed publicly on the Internet, but any user should not necessarily be able to perform any operation. A company's directory servicing its intranet can be stored behind a *firewall* to keep the general public from accessing it, but more security control may be needed within the intranet itself.

For example, any intranet user should be able to look up an employee's e-mail address, but only the employee themselves or a system administrator should be able to change it. Members of the personnel department might have permission to look up an employee's home telephone number, but their co-workers might not. Depending on the confidentiality of the data, information may need to be encrypted before being transmitted over the network. A security policy defines who has what type of access to what information, and is defined by the organization that maintains the directory.

A directory should support the basic capabilities needed to implement a security policy. The directory in this case is one of the components by which a security mechanism is put in place for the whole network. It is also one of the network resources that itself needs protecting.

First, a method is needed to authenticate users. Authentication verifies that users are who they say they are. A user name and password is a basic authentication scheme. Once users are authenticated, it must be determined if they have the authorization or permission to perform the requested operation on the specific object.

Authorization is often based on access control lists (ACLs). An ACL is a list of authorizations that may be attached to objects and attributes in the directory. An ACL lists what type of access each user or a group of users is allowed or denied. In order to make ACLs shorter and more manageable, users with the same access rights are often put into security groups. Table 7 shows an example ACL for an employee's directory entry.

Table 7. Example ACL for an employee's directory entry

User or Group	Access Rights
owner	read, modify (but not delete)
administrators	all
personnel	read all fields
all others	read restricted

The authentication and authorization methods and related standards and proposals relevant to LDAP, like *Secure Sockets Layer (SSL)* and *Simple Authentication and Security Layer (SASL)*, are discussed in more detail in 2.6, "Authentication and security services" on page 21.

#### G.1.4 Users, platforms, and networks

An understanding of users (what a user is, where the information about a user is stored, how it is used, and so on) is key to understanding the general relationship between white page directory services and platform operating systems.

Starting with a white page directory service, there are two obvious ways to think about users:

- A user may be an LDAP directory user. LDAP sessions begin with a bind operation. For cases other than unauthenticated access, a bind request must supply a distinguished name (DN) and a password or another type of credential. So, a user ID must be defined in the directory server (the user must have a DN) in order to be authenticated to access the directory.
- A user may have a white pages entry in an LDAP directory. The common white pages or phone book application will contain information about

people, in most cases many people. There can be, but does not have to be, a relationship between a white pages entry and an LDAP directory user entry; that is, everyone listed in the white pages directory may not have access to the directory. The inverse may also be true; people not listed in the directory may access the data as, for example, when internal users access a directory that stores customer person objects.

Most operating system platforms provide local or remote access for users; so, a user ID can also be handled in the following ways:

- A user may have a system-defined user ID. The user ID must be defined on the system platform in order to be authenticated so they can be a user of that platform for access to system services, such as file and print sharing. Such user IDs are usually accompanied by a password, or, with increasing importance, by a digital certificate signed by a known Certificate Authority (CA).

Historically, multi-user interactive applications and subsystems used the platform-level user definitions on the hosted platform to authenticate user access to transactions, programs and data. While single sign-on, and the overall security architecture are beyond the scope of this guide, it is sufficient to say that, as the number of systems in a network continues to grow and applications themselves are distributed across multiple systems, the strict application-to-host system user affinity is insufficient. So, many applications have developed their own private user and authentication files.

- A user may be an application user. This user ID must be defined/enrolled in the specific application in order to be authenticated and have access to that application and its resources.

It is important to understand this current topology, with its mix of semantics, syntax, and backing stores for user information, in order to understand the context of users within a distributed environment.

### **G.1.5 Directory versus database**

A directory is often described as a database, but it is a specialized database that has characteristics that set it apart from, for example, general purpose relational databases. One special characteristic of directories is that in general they are accessed (read or searched) much more often than they are updated (written). Hundreds of people might look up an individual's phone number, or thousands of print clients might look up the characteristics of a particular printer. But the phone number or printer characteristics rarely change.

Directories must be able to support high volumes of read requests, so they are typically optimized for read access. Write access might be limited to system administrators or to the owner of each piece of information. A general purpose database, on the other hand, needs to support applications such as airline reservations and banking with high update volumes.

Directories are meant to store relatively static information and are optimized for that purpose; they are not appropriate for storing information that changes rapidly. For example, the number of jobs currently in a print queue probably should not be stored in the directory entry for a printer because that information would have to be updated frequently to be accurate. Instead, the directory entry for the printer could contain the network address of a print server. The print server could be queried to learn the current queue length if desired. The information in the directory (the print server address) is static, whereas the number of jobs in the print queue is dynamic.

Another important difference between directories and general purpose databases is that directories may not support transactions. Transactions are all-or-nothing operations that must be completed in total or not at all. For example, when transferring money from one bank account to another, the money must be debited from one account and credited to the other account in a single transaction. If only half of this transaction completes or someone accesses the accounts while the money is in transit, the accounts will not balance. General-purpose databases usually support such transactions, which complicates their implementation.

Directories deal mostly with read requests, so the complexities of transactions can be avoided. If two people exchange offices, both of their directory entries need to be updated with new phone numbers, office locations, and so on. If one directory entry is updated, and then other directory entry is updated there is a brief period during which the directory will show that both people have the same phone number. Because updates are relatively rare, such anomalies are considered acceptable.

In contrast to directories, general-purpose databases must support arbitrary applications such as banking and inventory control, so they allow arbitrary collections of data to be stored. On the other hand directories may be limited in the type of data they allow to be stored (although the architecture does not impose such a limitation). For example, a directory specialized for customer contact information might be limited to storing only personal information such as names, addresses, and phone numbers. If a directory is extensible, it can be configured to store a variety of types of information, making it more useful to a variety of programs.

Another important difference between a directory and a general-purpose database is in the way information can be accessed. Most databases support a standardized, very powerful access method called Structured Query Language (SQL). SQL allows complex update and query functions at the cost of program size and application complexity. LDAP directories, on the other hand, use a simplified and optimized access protocol that can be used in slim and relatively simple applications.

Directories are not intended to provide as many functions as general-purpose databases. Thus they can be optimized to economically provide more applications with rapid access to directory data in large distributed environments. The intended use of directories is restricted to a read-mostly, non transactional environment, therefore both the directory client and directory server can be simplified and optimized.

#### What About the Future?

Many of the differences just mentioned may lead to the suspicion that a directory is no more than a limited-function database. This is indeed partly true, since one of the important design goals of a directory service is that it can be accessed and used from relatively small and simple applications. In fact, certain vendor products, such as IBM's SecureWay Directory, use a relational database under the cover to implement the functions, but directories do not specify which underlying database to use. Also, proposals are being discussed in the standards bodies to add some functions to future versions of LDAP that currently are specific to databases, such as support for transactional updates.

### G.1.6 Directory synchronization

*Directory synchronization* traditionally has been oriented toward messaging infrastructures, because this was where the early major requirements were formulated. We use this as an example to show how Directory Synchronization works.

Most e-mail systems today provide a directory that enables users to look up and select mail recipients. However, because mail systems have developed independently, a user of one type of mail system cannot look at entries in the directory of a different mail system.

This limitation, and the need for enterprises to form a unified messaging system, indicates that a mechanism for synchronizing name and address

information across heterogeneous LAN-based or mainframe-based mail systems is required.

The same structure can be used for enterprise directories, shown in Figure 71. In this example we have three different local directories: a departmental mail system, a LAN-based system and a mainframe system. The directory synchronization handles three steps:

- It sends a set of information from the local directory to the enterprise directory (uploading from client to server).
- Backbone synchronization between multiple enterprise directories occurs.
- The changes are sent from an enterprise directory to the local directories (downloading from server to client).

Because of this scheduled process it is possible to have all the necessary information about directory entries in all systems in the heterogeneous environment.

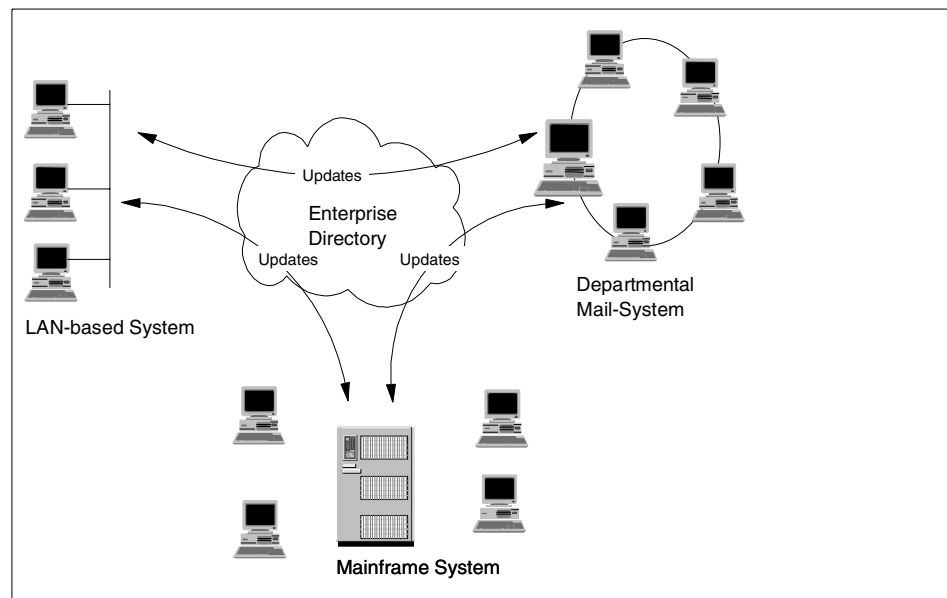


Figure 71. Directory synchronization

The process of Directory synchronization replicates directory information across a heterogeneous network. It make it possible to have informations in each directory which is used by the users in the enterprise network.

There are two common methods available to synchronize existing directories.

- Synchronization based on having the same user name and password in all directories (point-to-point synchronization or n-way synchronization)
- Integration of existing directories into one metadirectory (one way and two way synchronization)

The first method, based on the use of the same user name and password, requires that a synchronization mechanism exists between the different directory resources and types.

To do this, a common identifier (unique key) is required on all systems. This unique key can be as simple as a personal number, a social security number, or any other custom-made string, as long as it is unique and is supported by all systems. Based on the systems used, there will be cases where multiple instances of user credentials, as well as aliases, are required.

A typical set of user credentials would then look similar to the following attributes list, in the Domino Directory and the Windows NT Directory:

*Table 8. Typical user credentials*

<b>Domino Directory</b>	<b>Windows NT Directory</b>
User Name Short Name Network Account Name	User Name
User Name First Name Last Name	Full Name
Password	Password
Unique identifier	Unique identifier

Common attributes or attribute mapping tables will ensure that during the synchronization process between the systems all unique user information is synchronized. In a few cases, directory mapping tables will be used to make it possible for the end users to match certain attributes to custom defined fields.

An additional method uses a central directory to store all relevant information such as user names, passwords, and group names, as well as user certificates. This can be any directory store capable of storing all the required information and with access to the connecting systems and applications.

The challenge is to integrate all the existing systems and applications to authenticate through this central directory. Based on the available tools and

APIs, this can result in multiple prompts to the end user for authentication. There are currently some limitations related to the implementation of a common directory with a Notes/Domino environment.

The typical tools used for this will focus on application integration, such as:

- C/C++
- Visual Basic
- LDAP APIs
- Java
- ADSI (MS Active Directory Services Interface)
- JNDI (Java Naming and Directory Interface)
- LEI 3.0 (Lotus Enterprise Integrator)

---

## **G.2 Directory standards**

In the 1970s, the integration of communications and computing technologies led to the development of new communication technologies. Many of the proprietary systems that were developed were incompatible with other systems. It became apparent that standards were needed to allow equipment and systems from different vendors to interoperate. Two of the independent major standardizations efforts are LDAP and X.500.

### **G.2.1 X.500 - the Directory Service Standard**

During the 1980s, the growth in implementations of wide area network communication forced the deployment of a new set of networking protocols called open system interconnection (OSI). OSI presented a seven layer model of communications. Part of the standards developed by CCITT is a definition of generic directory service. CCITT defined the first X.500 standard in 1988, which then became ISO 9594, Data Communications Network Directory, Recommendations X.500/X.521 in 1990, though it is still commonly referred to as X.500.

Although the context and the content of both are the same, the two organizations use different terms: the ISO describes ISO/IEC 9594 as a multi-part standard, whereas the ITU-T refers to X.500 as a series of recommendations. To avoid confusion, we adopt the term "specification" which is accepted by both ISO and ITU-T.

The Directory Specifications are available in four editions:



1988 edition: This is the first edition and is issued as the multi-part standard ISO/IEC 9594:1990 and the CCITT X.500 (1988) Series of Recommendations. This edition specifies services, protocols and procedures necessary for basic directory operations, information models for how information is structured, and it specifies some commonly usable information objects. In addition, it provides a common framework for general authentication techniques.

1993 edition: This second edition is issued as ISO/IEC 9594: 1995 and as ITU-T X.500 (1993). Added Functions are, for example, shadowing of directory information, replication, access control and the expansion of the information model. Administrative capabilities are built in, too.

1997 edition: This edition adds a feature called contexts, which allows information to be distinguished according to the context in which it is being accessed. Another important addition is the provision of OSI management of the directory. It also has added and extended important security features.

2001 edition: This edition include higher internet (TCP/IP) integration.

Further extensions of the Directory specifications are currently in progress, and will include:

- The new service concept, which allows an administrator authority to tailor the service provided to the user through elaborate service administration tools.
- The hierarchical group feature, which allows establishment of a hierarchical structure independent of the DIT structure.
- The family of entities feature, which allows information about an object to be structured in a more logical way and allows powerful matching and rules for return of information.
- The mapping-based searches features, which allows search requests to be chances to provide a mapping between the world as the user sees it and the world as it is reflected by the directory.

For more information about the defined standards see Appendix B, “LDAP and X.500 Standards” on page 159.

X.500 organizes directory entries in a hierarchical name space capable of supporting large amounts of information. For example, in the general case, a name consists of several components reflecting hierarchy. This naming tree is called the directory information tree (DIT), as a directory entry is associated with each vertex of this tree, where the entry holds information about the

object having the corresponding name (see G.2.1.6, “The X.500 directory structure” on page 232).

The following sections explain the different models and protocols regarding X.500.

#### **G.2.1.1 The X.500 model**

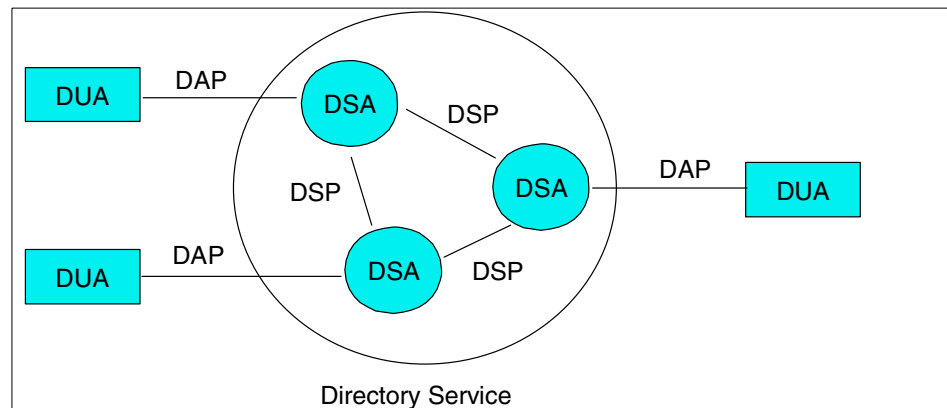
The X.500 directory system consists of three principal components:

- The Directory Information Base (DIB)
- Directory User Agents (DUAs)
- Directory Service Agents (DSAs)

If a user will get information stored in a directory they get access over a DUA, which represents the “client” side of the directory services. Users can be either people or application programs.

#### **G.2.1.2 The X.500 protocols**

To define the connection between the different directory functional components, two protocols were developed (see Figure 72).



*Figure 72. Distributed directory model*

The Directory Access Protocol (DAP) defines how DUAs get access to the information stored in DSAs.

If the information stored in the directory grows it might be necessary to split the DIB in multiple DSAs. A Directory Service Protocol (DSP) is used between two DSAs to query user information lookups over multiple DSAs.

A set of one or more DSAs and zero or more DUAs managed by a single organization may form a Directory Management Domain (DMD). A DMD may be an Administration DMD (ADDMD) or a Private DMD (PRDMD), depending on whether or not it is being operated by a public telecommunication organization or by service provider.

There are more protocols in X.500, but those are the basic ones.

### **G.2.1.3 Overview of the Directory Access Protocol (DAP)**

Additionally we have to discuss how the DSAs cooperate to provide a user service.

#### ***Bind and unbind operations***

The Directory bind operation is the operation which establishes a connection between the DUA and the DSA. Conversely, the Directory Unbind Operation is the operation which closes the connection between the two.

#### ***Interrogation operations***

There are four interrogation operations:

- The Read operation reads the information from one specific directory.
- The Compare operation compares the user-presented attribute value with those actually existing in the entry.
- The List operation lists the immediate subordinates of an entry.
- The Search operation is used to search portions of the DIT and to return selected information about selected entries.

In addition to these, the Abandon operation allows the user to abandon any of the above interrogation operations, if they are no longer interested in obtaining the result.

#### ***Modification operations***

Several different operations to modify entries in a directory were defined in the 88 standards; most of them were removed in the 93 edition of the standards:

- The AddEntry operation
- The RemoveEntry operation
- The ModifyRDN operation
- The ModifyDN operation
- The ModifyEntry operation

#### **G.2.1.4 Overview of the Directory System Protocol (DSP)**

This protocol is very similar to DAP. It is used between different DSAs to answer a user's DAP queries.

The first step must be the bind operation, which works in the same way as the DAP Bind Operation.

The operations that flow between the DSAs are called chained operations. Each of the DAP operations has a corresponding DSP chained operation, and each result has a corresponding chained result.

#### **G.2.1.5 Overview of the Directory Information Shadowing Protocol**

The Directory Information Shadowing Protocol (DISP) is used to transfer information from the shadow supplier to the shadow consumer. The DISP consists of three operations plus the DISP Shadow bind and Unbind operations, which are used to open and close connections. Either of two operations may be used to prepare both of the DSAs to send and receive the shadowed information. The third operation is used to actually transfer the shadowed data. Either the Coordinate Shadow Update operation, which is sent from the shadow supplier to the shadow consumer, or, alternatively, the Request Shadow Update operation, which is sent from the shadow consumer to the shadow supplier, may be used to synchronize the DSAs for the next batch of updates. The Update Shadow Operation is used to actually transfer the batch of updates from the supplier to the consumer. Each batch of updates is timestamped by the supplier, so that they can be uniquely identified. This aids discovery of the cause, in the event of a failure.

#### **G.2.1.6 The X.500 directory structure**

The information stored in the DIB is structured as a hierarchical information tree, called Directory Information Tree (DIT). Each level of the DIT is called an Object Class; all objects in the same class share the same attributes (see Figure 73 on page 233).

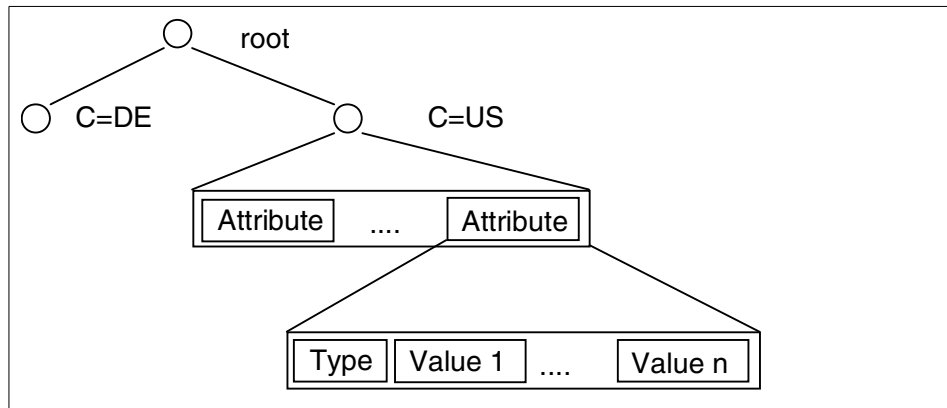


Figure 73. Directory entry structure

A DIT address is made up of a sequence of Relative Distinguished Names (RDNs), which are ordered sequences of attributes.

The example shown in Figure 74 explains the structure of DITs and the special terms.

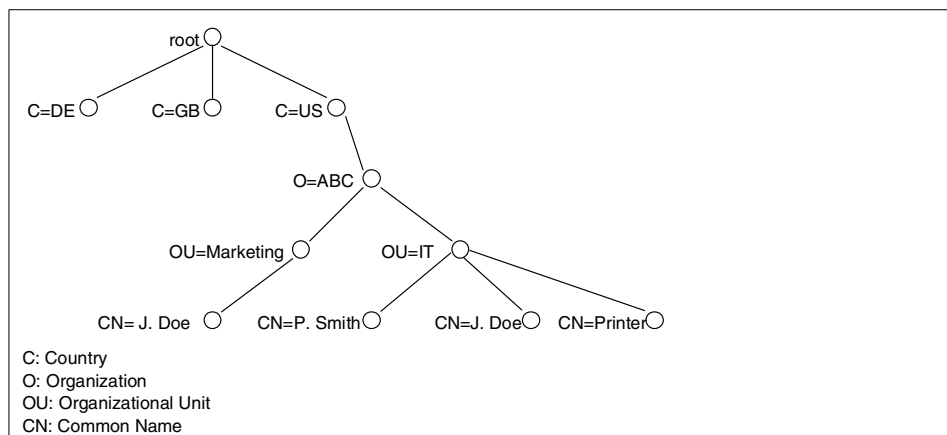


Figure 74. Example of a directory information tree

The example shows a DIT with entries for three countries. The name space for US is divided into an organizational level name space with different user levels.

If you look at the hierarchy for the entry of P. Smith you can see the following naming attributes, which represents the following distinguished Name:

- Country: US
- Organization: ABC
- Organizational Unit: IT
- Common Name: P.Smith

There are two entries with the common name J.Doe, but they can be distinguished by their full RDNs:

- {C=US, O=ABC, OU=Marketing, CN=J. Doe}
- {C=US, O=ABC, OU=IT, CN=J. Doe}

This shows you that the directory requires the full distinguished name to locate an entry.

Users typically do not need to type in the full name. Software solutions for DUA can expand it automatically to the full name and provide selections among similar entries.

Different vendors use different structures of DITs. For example, Microsoft Active Directory uses domain component (dc) instead of country (c) and organization (o). Some deployments of DITs prefer to use DNS-based naming, which is available with most directory products. You can find more information in 7.2, “Integrating with other directories and applications” on page 110.

#### **G.2.1.7 Operation of the X.500 model**

The DUA interacts with the directory by communicating with one or more DSAs. There are different ways of request-handling, as illustrated in Figure 75 through Figure 79.

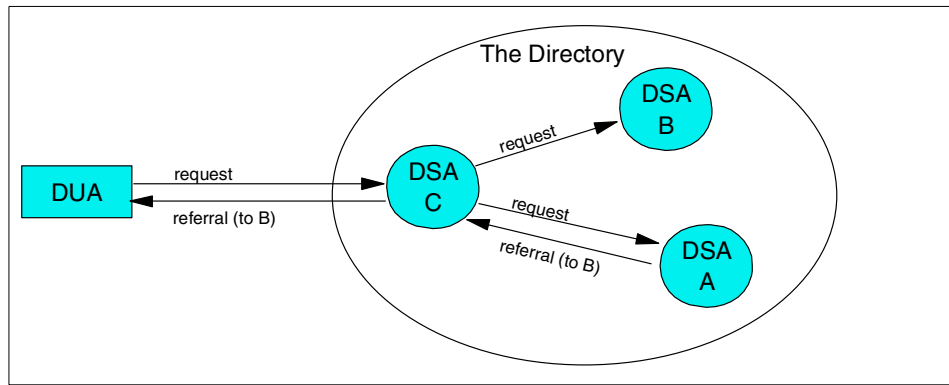


Figure 75. Referrals a

The referrals from DSA A will be received by DSA C, which is responsible for every conveying to the DSA B or conveying the referral to the originating DUA.

If DSA C returns the referral to the DUA, the request to DSA B will not occur.

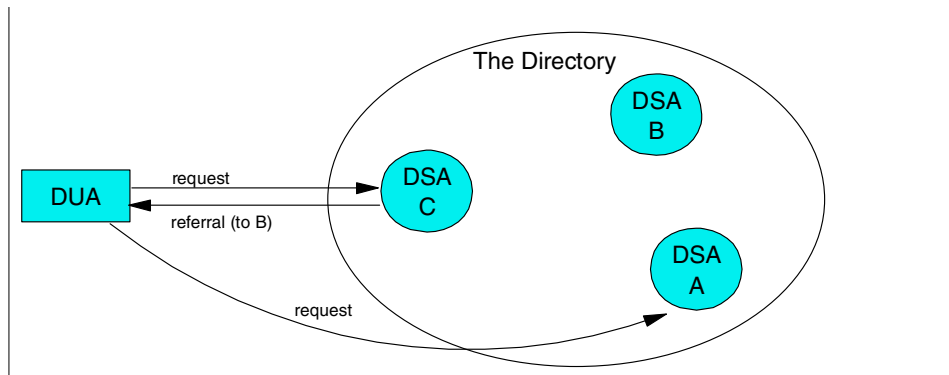


Figure 76. Referrals b

Referrals provide a way for servers to refer clients to additional directory servers. With referrals you can:

- Distribute namespace information among multiple servers
- Provide knowledge of where data resides within a set of interrelated servers
- Route client requests to the appropriate server

Using referrals has several advantages. Referrals let you:

- Distribute processing overhead, providing primitive load balancing
- Distribute administration of data along organizational boundaries
- Provide potential for widespread interconnection, beyond an organization's own boundaries

In the case shown in Figure 76, the DUA receives the referral from DSA C and is responsible for reissuing the request directly to DSA A (which is named in the referral from DSA C).

If a DUA is connected directly to one DSA and this DSA cannot respond, it will send a request to another DSA. This process is called *chaining*.

It is possible to pass the request through several DSAs before response is returned. This is shown in Figure 77.

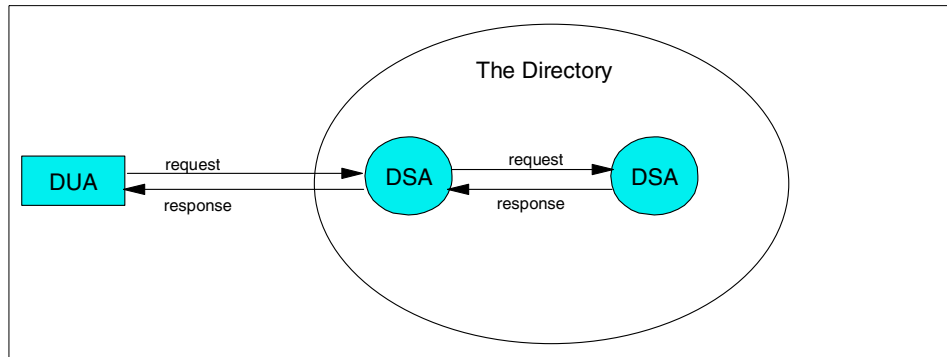


Figure 77. Uni-chaining

Forwarded the request to two or more DSAs is called multi-chaining; this is illustrated in Figure 78 on page 237.



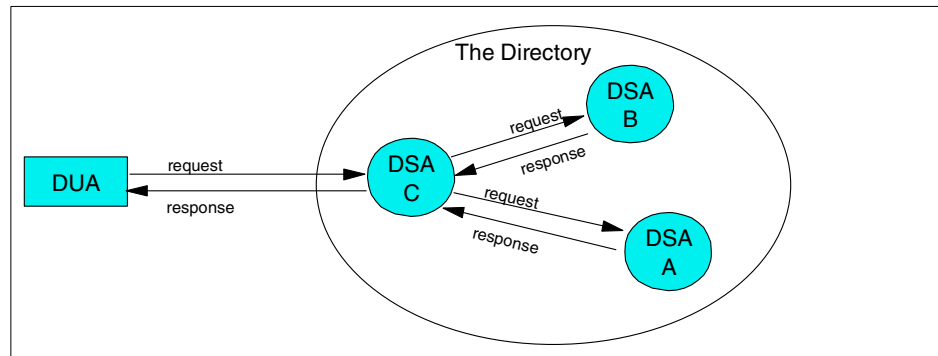


Figure 78. Multi-chaining

In a directory environment you can find all the approaches. A hybrid approach that combines functional interactions may be needed to satisfy a request. An example is shown in Figure 79.

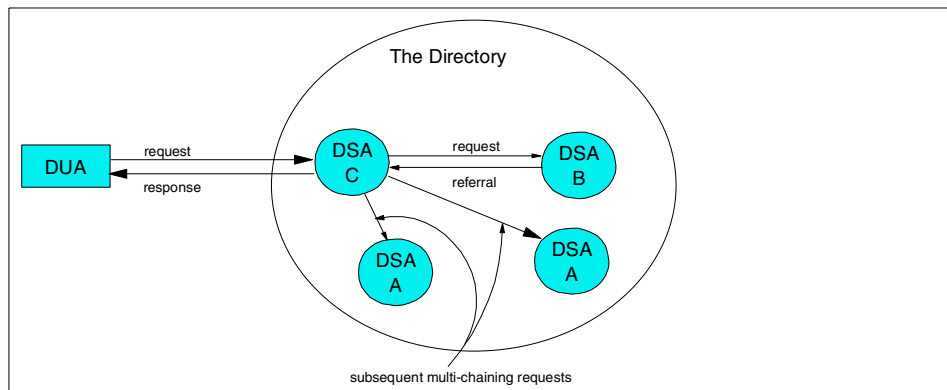


Figure 79. Mixed modes hybrid approach

## G.2.2 X.509 certificates

Although there have been several proposed formats for public key certificates, most commercial certificates available today are based on the international standard ITU-T Recommendation X.509 (formerly CCITT X.509).

X.509 certificates are used in secure Internet protocols like:

- Secure Socket Layer (SSL)
- Secure Multi-purpose Internet Mail Extension (S/MIME)

- Secure Electronic Transaction (SET)

#### **G.2.2.1 The X.509 standard**

Originally, the X.509 standard was intended to specify the authentication service for X.500 directories. Directory authentication in X.509 can be done using either secret-key techniques or public-key techniques. The latter is based on public-key certificates. At present, the public-key certificate format defined in the X.509 standard is widely used and supported by a number of protocols in the Internet world. X.509 standard does not specify a particular cryptographic algorithm; however, apparently RSA algorithm is the most broadly used one.

The initial version of X.509 was published in 1988. The public-key certificate format defined in this standard is called X.509 version 1 (X.509v1). When X.500 was revised in 1993, two more fields were added, resulting in the X.509 version 2 (X.509v2) format.

X.509 version 3 (X.509v3) was proposed in 1994. X.509v3 extends v2 in order to address some of the security concerns and limited flexibility that were issues in versions 1 and 2. The major difference between versions 2 and 3 is the addition of the extensions field. This field grants more flexibility since it can convey additional information beyond just the key and name binding. In June 1996, standardization of the basic v3 format was completed.

#### **G.2.2.2 X.509 certificate content**

An X.509 certificate consists of the following fields:

- Version of certificate format
- Certificate serial number
- Digital signature algorithm identifier (for issuer's digital signature)
- Issuer name (that is, the name of the Certification Authority)
- Validity period Subject (that is, user or server) name
- Subject public-key information: algorithm identifier and public-key value
- Issuer unique identifier, version 2 and 3 only (added by version 2)
- Subject unique identifier, version 2 and 3 only (added by version 2)
- Extensions, version 3 only (added by version 3)
- Digital signature by issuer on the above fields

Standard extensions include subject and issuer attributes, certification policy information, and key usage restrictions, among others.

### G.2.3 LDAP

X.500 is based on the client/server model of distributed computing and LDAP represents the client implementation of that model. LDAP evolved as a lightweight protocol for accessing information in X.500 directory services. It has since become independent of X.500; servers that specifically support the LDAP protocol rather than the X.500 Directory Access Protocol (DAP) are now common. The success of LDAP has been largely due to the following characteristics that make it simpler to implement and use, compared to X.500 and DAP:

- LDAP runs over TCP/IP rather than the OSI protocol stack. TCP/IP is less resource-intensive and is much more widely available, especially on desktop systems.
- The functional model of LDAP is simpler. It omits duplicate, rarely-used and esoteric features. This makes LDAP easier to understand and to implement.
- LDAP uses strings to represent data rather than complicated structured syntaxes such as ASN.1 (Abstract Syntax Notation One).

Several standards in the form of IETF RFCs exist for LDAP. The following is a list of the core RFCs that apply for LDAP Version 2 and Version 3:

#### LDAP Version 2:

- RFC 1777: Defines the LDAP protocol
- RFC 1778: The String Representation of Standard Attribute Syntax
- RFC 1779: A String Representation of Distinguished Names
- RFC 1959: An LDAP URL Format
- RFC 1960: A String Representation of LDAP Search Filters

#### LDAP Version 3:

- RFC 2251: LDAP protocol - V3
- RFC 2252: Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
- RFC 2253: Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
- RFC 2254: The String Representation of LDAP Search Filters
- RFC 2255: The LDAP URL Format
- RFC 2256: A Summary of the X.500 (96) User Schema for use with LDAPv3

More information and a brief description of these standards (RFCs) can be found in Appendix B, "LDAP and X.500 Standards" on page 159.

In this book, the term LDAP generally refers to LDAP Version 3. Differences between LDAP Version 2 and LDAP Version 3 are noted when necessary.

### G.2.3.1 LDAP architecture

LDAP defines the content of messages exchanged between an LDAP client and an LDAP server. The messages specify the operations requested by the client (search, modify, delete, and so on), the responses from the server, and the format of data carried in the messages. LDAP messages are carried over TCP/IP, a connection-oriented protocol; so there are also operations to establish and disconnect a session between the client and server.

However, for the designer of an LDAP-based directory, it is not so much the structure of the messages being sent and received over the wire that is of interest. What is important is the logical model that is defined by these messages and data types, how the directory is organized, what operations are possible, how information is protected, and so forth.

The general interaction between an LDAP client and an LDAP server takes the following form:

- The client establishes a session with an LDAP server. This is known as *binding* to the server. The client specifies the host name or IP address and TCP/IP port number where the LDAP server is listening. The client can provide a user name and a password to properly authenticate with the server. Or the client can establish an anonymous session with default access rights. The client and server can also establish a session that uses stronger security methods such as encryption of data.
- The client then performs operations on directory data. LDAP offers both read and update capabilities. This allows directory information to be managed as well as queried. LDAP also supports searching the directory for data meeting arbitrary user-specified criteria. Searching is a very common operation in LDAP. A user can specify what part of the directory to search and what information to return. A search filter that uses Boolean conditions specifies what directory data matches the search.
- When the client is finished making requests, it closes the session with the server. This is also known as *unbinding*.

Although it is not defined by the LDAP protocol itself, there is a well-known LDAP API (application program interface) that allows applications to easily interact with LDAP servers. The API can be considered an extension to the LDAP architecture. The C language LDAP API associated with LDAPv2 is an informational RFC. The update to it has not yet progressed to RFC status. However it has achieved *de facto* standard status because it is supported by

all major LDAP vendors. The philosophy of the LDAP API is to keep simple things simple. This means that adding directory support to existing applications can be done with low overhead. As we will see in 6.3, “Using application development tools with Domino Directory services” on page 105, this interface is reasonably easy to use and implement in applications.

Because LDAP was originally intended as a lightweight alternative to DAP for accessing X.500 directories, it follows an X.500 model (see G.2.1, “X.500 - the Directory Service Standard” on page 228). The directory stores and organizes data structures known as *entries*.

A directory entry describes some object. An object class is a general description of an object as opposed to the description of a particular object. For instance, the object class *person* has a *surname* attribute, whereas the object describing John Smith has a *surname* attribute with the value Smith. The object classes that a directory server can store and the attributes they contain are described by schema. Schema define what object classes are allowed where in the directory, what attributes they must contain, what attributes are optional, and the syntax of each attribute. For example, a schema could define a *person* object class. The *person* schema might require that a *person* have a *surname* attribute that is a character string, specify that a *person* entry can optionally have a *telephoneNumber* attribute that is a string of numbers with spaces and hyphens, and so on.

LDAP defines operations for accessing and modifying directory entries such as:

- Searching for entries meeting user-specified criteria
- Adding an entry
- Deleting an entry
- Modifying an entry
- Modifying the distinguished name or relative distinguished name of an entry (move)
- Comparing an entry

Objects can be derived from other objects. This is known as subclassing. For example, suppose an object called *person* was defined that included a *surname* and so on. An object class *organizationalPerson* could be defined as a subclass of the *person* object class. The *organizationalPerson* object class would have the same attributes as the *person* object class and could add other attributes such as *title* and *officenum*. The *person* object class would be called the superior of the *organizationalPerson* object class. One

special object class, called top, has no superiors. The top object class includes the mandatory *objectClass* attribute. Attributes in top appear in all directory entries as specified (required or optional).

Each directory entry has a special attribute called *objectClass*. The value of the *objectClass* attribute is a list of two or more schema names. These schema define what type of object(s) the entry represents. One of the values must be either top or alias. Alias is used if the entry is an alias for another entry, otherwise top is used. The *objectClass* attribute determines what attributes the entry must and may have.

The special object class *extensibleObject* allows any attribute to be stored in the entry. This can be more convenient than defining a new object class to add a special attribute to a few entries, but also opens up that object to be able to contain anything (which might not be a good thing in a structured system).

### G.2.3.2 LDAP models

LDAP can be better understood by considering the four models upon which it is based:

- Information** Describes the structure of information stored in an LDAP directory.
- Naming** Describes how information in an LDAP directory is organized and identified.
- Functional** Describes what operations can be performed on the information stored in an LDAP directory.
- Security** Describes how the information in an LDAP directory can be protected from unauthorized access.

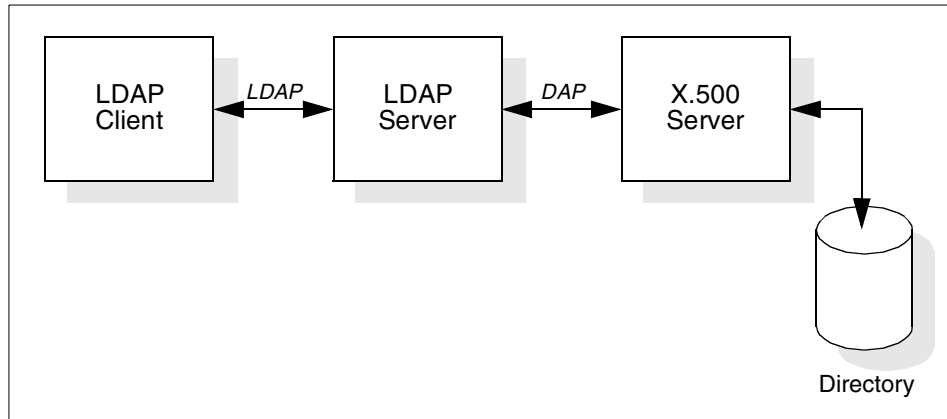
For more information about LDAP Models see *Understanding LDAP*, SG24-4986.

### G.2.3.3 LDAP: protocol or directory

LDAP defines a communication protocol. That is, it defines the format of messages used by a client to access data in a directory service that listens for and responds to LDAP requests. LDAP does not define the directory service itself. Yet people often talk about LDAP directories. Others say LDAP is only a protocol, that there is no such thing as an LDAP directory. What is an LDAP directory?

An application client program initiates an LDAP message by calling an LDAP API. But an X.500 directory server does not understand LDAP messages.

The LDAP client actually communicates with a gateway process (also called a proxy or front end) that forwards requests to the X.500 directory server (see Figure 80). This gateway is known as an LDAP server. It services requests from the LDAP client. It does this by becoming a client of the X.500 server.



*Figure 80. LDAP server acting as a gateway to an X.500 server*

As the use of LDAP grew and its benefits became apparent, people who did not have X.500 servers or the environments to support them wanted to build directories that could be accessed by LDAP clients. So, why not have the LDAP server store and access the directory itself instead of only acting as a gateway to X.500 servers, as shown in Figure 81? This eliminates any need for the OSI protocol stack. Of course this makes the LDAP server much more complicated since it must store and retrieve directory entries. These LDAP servers are often called stand-alone LDAP servers because they do not depend on an X.500 directory server. Since LDAP does not support all X.500 capabilities, a stand-alone LDAP server only needs to support those capabilities required by LDAP.

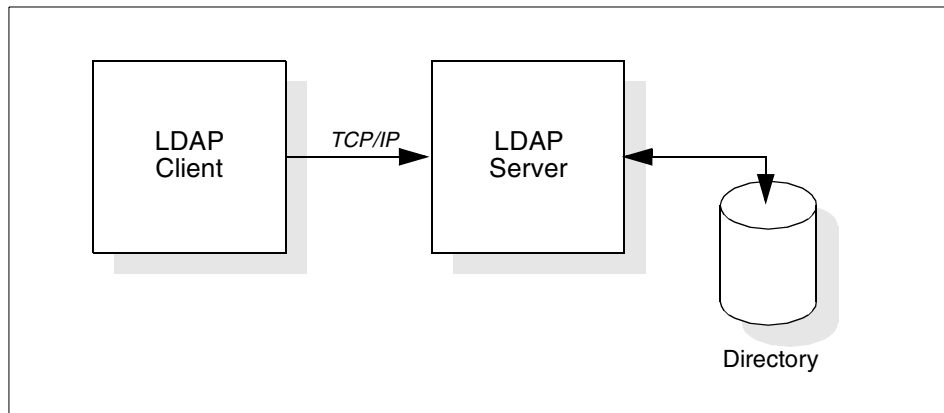


Figure 81. Stand-alone LDAP server

RFC 1777 (LDAP Version 2) discusses providing access to the X.500 directory. RFC 2251 (LDAP Version 3) discusses providing access to directories supporting the X.500 model. This change in language reflects the idea that an LDAP server can implement the directory itself or can be a gateway to an X.500 directory.

From the client's point of view, any server that implements the LDAP protocol is an LDAP directory server, whether the LDAP server is one that supports the X.500 model or is a gateway to an X.500 server.

The directory that is accessed can be called an LDAP-based directory, whether the directory is implemented by a stand-alone LDAP server or by an X.500 server.

---

### G.3 Enterprise directory

The "enterprise directory" within an organization is classically envisioned to be the authoritative repository of heterogeneous corporate information that can be used by a variety of applications. While this definition has been relatively consistent over time, the technology associated with it has changed.

The initial promise of enterprise directory functionality was based on an anticipated widespread adoption of the X.500 directory standard. More recently with the emergence of LDAP as the de facto commercial standard for directory services, the IETF and related committees have focused on



ensuring that this protocol has the characteristics needed to be effectively used in this central infrastructure role.

The features required of an enterprise directory include standards implementation, extensibility, manageability, security and scalability.

The current standard of choice is LDAP. Its latest iteration is LDAP v3. Major industry participants, including Lotus, IBM, Microsoft, Oracle, Novell, Sun/Netscape, Nexor, Siemens, Computer Associates and others, have embraced it and have placed LDAP-based directory offerings on the market. Also important is the industry's efforts to ensure that these standards-based implementations are interoperable. For example, an industry forum called DirConnect periodically hosts technical representatives from various directory vendors to incorporate their products. They spend several days identifying and correcting interoperability issues.

Extensibility is the ability of a directory product to adapt to the requirements of the organization it is supporting. This is generally discussed in terms of extensions to the LDAP schema. This is a set of rules which define the nature of the information items and how they interact. In an enterprise environment, this set of directory-defining rules must be not only extensible, but also capable of being accessed by applications seeking to use the information stored in the directory. The LDAP standards provide a mechanism for defining and publishing the initial schema and for extending that schema if necessary. So an enterprise directory should not only comply with the LDAP standards for populating, locating, and returning information in the directory, but also define and extend the directory schema.

Manageability refers to the capability of an organization to maintain control over the contents of the directory, its directory's performance and its ability to monitor, troubleshoot, and correct the various directory processes in use. Here again, the implementation of a standards-based directory can provide concurrent flexibility in the tools used to manage that directory and the enterprise directory environment.

Security is generally viewed in the context of maintaining control over access to the information stored in the directory. Since an enterprise directory is the authoritative repository of enterprise information, it is critical to maintain the ability to tightly control which specific individuals and groups are permitted to create, modify, and delete information in it. Furthermore, it is important to provide a granular level of control. This can range from a binary "have access or don't have access" (where if an individual has access to the directory they are free to impact anything) on one hand, to controls imposed at the individual attribute level. Currently there is no component of the LDAP

standard that addresses access control, so each vendor has implemented their own mechanisms. However, such a standard is being developed and should begin to appear in products during 2001.

Scalability of a directory is the least standard-dependent and most vendor-specific area of enterprise directory technology. Vendors have taken different approaches to achieving enterprise scalability for their products. Enterprises need to exercise caution, as they do in any performance measurement, to ensure that they understand two critical things when judging enterprise directory scalability. The first of these is to understand how the vendor's scalability claims are developed and supported, in order to understand how well that environment matches their own. The second is to have a realistic understanding of their own scalability needs. This is not an area where getting the biggest is necessarily the best. A good match between requirements and product capabilities, with a sufficient margin for realistic growth, is the most effective solution.

A final aspect of an organization's enterprise directory, just as critical as the others, is interoperability. While the definition of an enterprise directory implies a single directory construct, the reality for the vast majority of enterprises is that they will continue to function in a multi-directory environment for the near- to mid-term. So the ability to interoperate with the other directory stores in the organization is a key characteristic of an enterprise directory. Classically this requirement is met either through a directory synchronization mechanism or through metadirectory technology.

---

## **G.4 Metadirectory**

There are several interpretations of the term metadirectory, but it is usually used to describe a central enterprise directory that is able to synchronize with multiple directory systems and manage the relationship between heterogeneous namespaces. The term and the concept were originally scoped in a report entitled "Directory Services Strategic Overview: Meta-Directory Services" produced by the Burton Group in February 1996, and revised in 1998.

Despite the growing recognition of the common requirement for metadirectory solutions, there are as yet no standards for metadirectory, or any unequivocal consensus amongst vendors as to what exactly constitutes a metadirectory. Until a definition is achieved, metadirectory will remain a conceptual term based upon a broad set of user requirements for an integrated, distributed directory service, and used to describe a range of products, services and environments. In general, metadirectory products are batch or event-driven

middleware providing management capabilities across disparate directory and database systems; and a metadirectory deployment usually provides a co-ordinated administration system and a single logical namespace across a heterogeneous networking environment..

A metadirectory can be based on an existing directory product, or use an independent engine, such as X.500. In general, it is a trend amongst vendors to ensure that the "meta layer" management tools and directory synchronization engine are directory agnostic, and will work with any underlying directory, often using LDAP. Although the term is confused and often used interchangeably with "enterprise directory", not all metadirectory products require co-existence with any particular directory system: they may or may not use an independent directory. In other words, the actual repository used to store metadirectory-related information is not that significant.

Metadirectory is a directory-enabled application, where the application is the management of heterogeneous directory systems, and the role of the associated directory is primarily to store configuration and mapping data, but not necessarily data that is vital for the role of an enterprise directory. The benefits of an effective metadirectory solution is that it is an enabler of single point of administration and single sign-on, and may also be used to provide:

- lower administrative costs by acting as the source for consolidated directory information
- an authoritative source for public key certificates
- a central point for policy information
- efficient access directory information for new web based applications

Because metadirectory is a management tool, the directory product associated with the metadirectory solution may also be used as a company's enterprise directory, or it may exist as a standalone application.

#### **G.4.1 Metadirectory Systems**

As shown in Figure 82, a metadirectory system may consolidate either:

- a superset of data derived from multiple different repositories, with different different attributes managed from different systems; or
- a subset of data, maintaining the enterprise's common namespace alongside namespace mappings to all the connected systems, as well as one or two other key attributes, such as email and public key

Either approach is valid, and they depend very much on a company's strategy for directory use and application development.

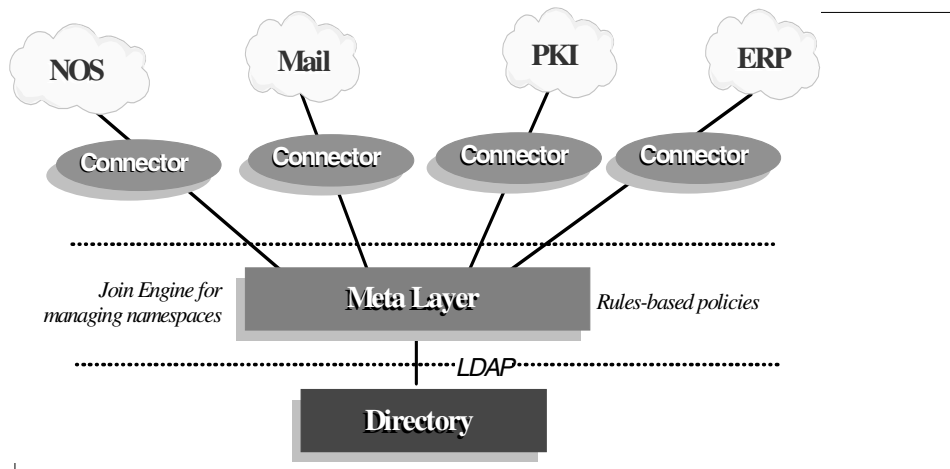


Figure 82. A metadirectory architecture

### G.4.2 Metadirectory Product Architecture

As shown in Figure 83 on page 249, typical metadirectory implementations include:

- Connectors: synchronization agents to extract and/or change information in source and destination repositories. Most products have connectors for relational databases, NOS, e-mail and other application directories.
- Join Engine: a central management process to direct the agents and transform data. This engine implements business rules describing how data is accessed and by whom.
- Optionally, a central LDAP-accessible directory.

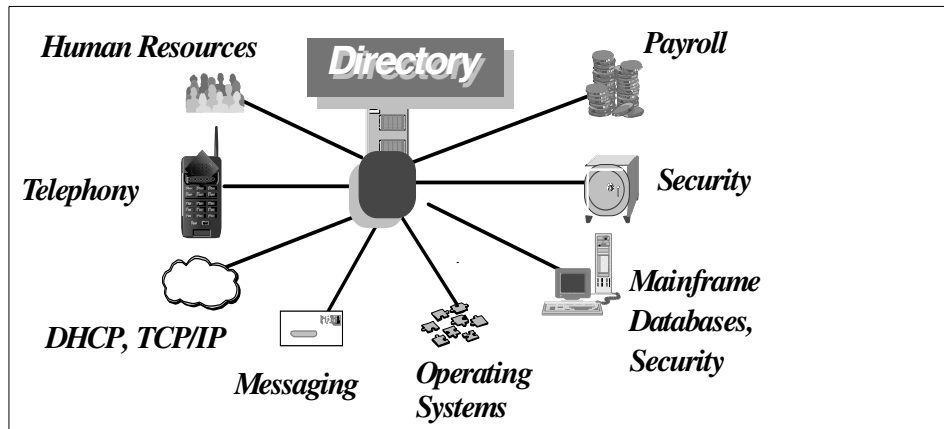


Figure 83. An enterprise metadirectory

Most implementations also include:

- A scheduler to manage the operational flow of the join engine and connectors.
- Scripting tools, either proprietary or based on standard scripting languages, to start auxiliary processes and provide programmatic control of both the join engine and connectors.
- A graphical or web user interface to manage objects, configure directory views and access results.

## G.5 Emerging Trends

Let us take a look at some trends in the directory standards arena.

### G.5.1 Converging on standards -- LDAP

Having published LDAPv2 as RFCs 1777-78 in 1995, and LDAPv3 as RFCs 2251-56 in December 1997, the two current IETF working groups dedicated to furthering the scope of LDAP-related directory functionality. The LDAP Extension (LDAPv3) working group defines and standardizes extensions to the LDAPv3 protocol and extensions to the use of LDAP on the Internet. The planned extensions include the following areas:

- Access Controls
- Server-side sorting of search results/Paged retrieval of search results
- Language tags

- Dynamic directories
- Referral and knowledge reference maintenance
- LDAP server discovery
- LDAP APIs
- CLDAP
- Signed directory information

Other areas such as deployment and schema definition and review will be handled by other groups if and when they turn out to be necessary for the deployment of LDAP and feasible for the group to tackle.

The second group, LDUP, addresses the issues of replication of data across servers running different implementations which is becoming an important part of providing a distributed directory service as LDAPv3 becomes more widely deployed. The LDAPv3 community to date has focused on standardizing the client-server access protocol, and so the LDUP group is chartered to standardize LDAPv3-based replication according to the following two models:

- **Multi-Master Replication:** entries can be written and updated on any of several replica copies, without requiring communication with other masters before the write or update is performed.
- **Master-Slave, or Single-Master Replication:** assumes only one server (the single-master) is allowed write-access to replicated data. (Note that master-slave replication can be considered a proper subset of multi-master replication.)

The new replication architecture will support both these approaches.

The LDUP working group first developed a set of requirements for LDAPv3 directory replication, and then wrote an applicability statement defining scenarios on which replication requirements are based. Six areas of working group focus have been identified, each leading to one or more documents to be published:

- LDAPv3 Replication Architecture
- LDAPv3 Replication Information Model
- LDAPv3 Replication Information Transport Protocol
- LDAPv3 Mandatory Replica Management
- LDAPv3 Update Reconciliation Procedures
- LDAPv3 Profiles

It is anticipated that consensus on the LDUP work will be reached during 2001, with the anticipated uptake by vendors over the following two years.

### **G.5.2 Leveraging directory services -- Directory Enabled Networks**

The Directory Enabled Network (DEN) specification is designed to provide the building blocks for more intelligent networks by mapping users to services, and mapping business criteria to the delivery of network services. This will enable applications and services to transparently leverage network infrastructure on behalf of the user, empower end-to-end services, and support distributed network-wide service creation, provisioning and management.

DEN defines a directory as a centralized repository that defines the relationship of users and applications to network services. DEN builds intelligent networks and networked applications that are managed holistically by associating users and applications to network services and according to a consistent and rational set of policies. This will result in a generation of cross-domain network applications and services that are intelligent and self-managing.

DEN builds upon the Distributed Management Task Force's (DMTF - <http://www.dmtf.org>) Common Information Model (CIM) standard to model functionality and management of network elements. DEN enables a company to manage its network as a single system and provides interoperability, data sharing, and transparency of the data source for cross-domain solutions.

The initial LDAP mappings for the CIM Core Schema as well as the CIM Schema v2.3, which now includes the User, Support, and Diagnostic models were completed in March 2000. The release of the LDAP mapping for the CIM Core and Physical Schema, as well as the CIM to LDAP mapping guidelines document were completed in June

2000. Additional mapping specifications are scheduled to be released by the end of 2000. The completion of this work should open the way for the vendor community to provide DEN-based products.

### **G.5.3 Common data definition -- DSML**

Directories typically store and manage information about each user in an enterprise - including names, addresses, phone numbers and access rights. Directories are increasingly storing metadata about available Web services, what they do, what they require for inputs, how to execute them, what the results will be, who wrote them and how to pay for them. Combined with the

power of XML - eXtensible Markup Language, the Internet's lingua franca for e-business - this information enables whole new classes of individually tailored applications for e-commerce. The definition of the XML schema for describing directory structure and data is DSML - Directory Services Markup Language.

Applications consume DSML documents as they would XML because DSML is a subset of XML. Applications can transmit DSML documents to other DSML-enabled applications on the Internet. This process effectively extends LDAP across firewalls to any Internet transport protocol, such as HTTP, FTP or SMTP, which is a major benefit for business-to-business (B2B) efforts.

DSML is also a major step forward in facilitating interoperability between different vendors' products, by describing their contents in XML. DSML-compliant directories can publish schema information as an XML document which can then be shared by other directories or applications. For example, account information can be maintained across multiple business partners, regardless of the underlying directory structure on each partner's site.

By leveraging the XML/DSML standards, applications can be enabled to react quickly to the needs of business, while leveraging a solid foundation of interoperability with backend systems. The combination of LDAP and XML provides data in a way that allows easy integration within both new and existing applications: whilst LDAP provides a means for accessing directory information, DSML provides the means for reading and understanding directory content in XML. So, DSML provides a standard for creating XML documents from the information that LDAP delivers.

A group of commercial vendors, comprising AOL (represented by Netscape), Bowstreet, IBM, Microsoft, Novell, Oracle and Sun came together in July 1999 to announce the DSML Forum, and following the release of DSML 1.0 have approached OASIS, the Organization for the Advancement of the Structured Information Standards (<http://www.oasis.org>), to charter a Technical Committee to enhance and refine the DSML standard, and to continue development of DSML in an open forum. All of the original DSML founding member organizations have already joined the process, as well as numerous DSML supporting members and new interested parties, including Lotus.



---

## Appendix H. Directory Schema

Schema is a core feature of any directory service describing how the directory structures the information it provides access to. Schema elements include object classes and attributes, which are represented in Domino as forms and subforms. As a rule most directory systems, including Notes/Domino, have determined their own schema elements, which in the past has led to irreconcilable interoperability problems. The standards have strived to help, but were presented with a conundrum: for example, X.500 defines a relatively small number of core schema elements, but leaves the further adoption and extension of schema to vendors and users in order to allow for maximum flexibility. This is both a blessing and a curse, as it means that whilst users are able to develop or adapt schema to their own particular requirements, it also by necessity implies that there is little convergence across the industry.

Schema elements are described in various X.500 documents from 1988 onwards and updated in later versions. In addition, the COSINE project PARADISE, the international X.500 pilot from the early nineties, defined in RFC1274 an additional set of attributes to be used in the “real world” of an X.500 deployment. The last of the defining documents pertaining to LDAPv3, RFC2256, entitled “A Summary of the X.500(96) User Schema for use with LDAPv3” and published in December 1997, summarized the work that had gone before for use by implementations of LDAPv3. These include:

- 1988: X.520
- 1993: X.501, X.509, X.520/521
- 1996: X.509, X.520

as well as RFC1274 and some new definitions.

As with X.500, RFC2256 was intended as a baseline for implementers, but it gave no guidance for schema elements that went beyond the core set defined in the IETF document. In November 1997, a group of vendor representatives meeting with the NAC (Network Applications Consortium) in Boston agreed on a core set of 37 schema elements considered necessary to describe an “Internet person”. The intention of this initiative was that any LDAP-enabled client would be able to find all the elements typically included on an Internet person’s business card (plus a number of administrative and security elements), irrespective of which LDAP server was being accessed. All eight vendors present (Banyan, IBM, Lotus, Microsoft, Netscape, Novell, Worldtalk, Zoomit) agreed to support what became known as the *Lightweight Internet Person Schema (LIPS)* in their products, and to make public this commitment.

There was further agreement that, if any one or any group of vendors should wish to extend the scope of the LIPS schema, this could be achieved by a two-thirds majority. Although the original LIPS work perseveres, the spirit of the agreement has been superseded by other initiatives such as the Directory Interoperability Forum and others.

Concurrently with the publication of the LIPS Declaration, work was going on to extend the scope of RFC2256, and, after a number of red herrings, the most widely adopted common Internet schema in circulation is the inetOrgPerson object class defined in RFC2798.

The stated purpose of RFC2798 is that:

The inetOrgPerson object class is a general purpose object class that holds attributes about people. The attributes it holds were chosen to accommodate information requirements found in typical Internet and intranet directory service deployments. The inetOrgPerson object class is designed to be used within directory services based on LDAP and the X.500 family of protocols, and it should be useful in other contexts as well. There is no requirement for directory services implementers to use the inetOrgPerson object class; it is simply presented as well-documented class that implementers can choose to use if they find it useful.

Like its predecessors, RFC2798 makes some new schema additions, but also pulls together schema elements from RFC1274, RFC2079 and RFC2256. It is now used as a baseline by most, but not all popular commercial products. The significant difference between LIPS and the schema definitions mentioned above is that LIPS attempts to clarify what schema elements are necessary to create the “Internet directory business card”, but stops short at identifying syntax or object identifiers. Most, but not all, of the 37 elements identified in LIPS can be found in inetOrgPerson.

Individual vendors, such as Lotus, when choosing an LDAP schema to base their service on created an object class which is a subset of RFC2256 and inetOrgPerson, but also includes further elements corresponding to the LIPS list. As such, the schema decision for most user organizations is determined by their choice of directory vendor (i.e., what is available out-of-the-box) and/or their decision to either extend the existing object classes or create their own. This second option is based on how extensively different the user organization's schema requirements are.

In conclusion, the choice of schema is ultimately determined by the user organization's requirement set, and to some extent is predicated by the choice of product. LIPS outlines the requirement for a person object class, and inetOrgPerson comes the closest to matching it.

The tables in this section show the correspondences between the different schema approaches mentioned above, and finally as adopted by the Domino Directory. It should be noted that the following attributes only represent a small subset of the full Domino LDAP schema (see Table 9).

Table 9. Domino OID (object identifier) allocation

Notes: 2.16.840.1.113678.2.2.2.2.xxx		
Object Classes	Domino OID xxx Numbers	Attribute Totals
Group	8, 9, 11-15	7
Person	4, 5, 7, 9, 16, 18-31, 71-96, 464	46
GeneralInfo	3, 5, 32-70	41
Server	108-427, 451-472	342

In Domino, the object classes for *person* follow the following hierarchy:

top (RFC2256)

    person (RFC2256)

        organizationalPerson (RFC2256)

            inetOrgPerson (RFC2798)

                ePerson (IBM elements)

                    dominoPerson (Lotus elements)

Consequently, Domino is able to support elements inherited from inetOrgPerson as well as supplementary attributes belonging to LIPS in the dominoPerson object class.

Figure 84 on page 256 shows the LDAP object classes in Domino.

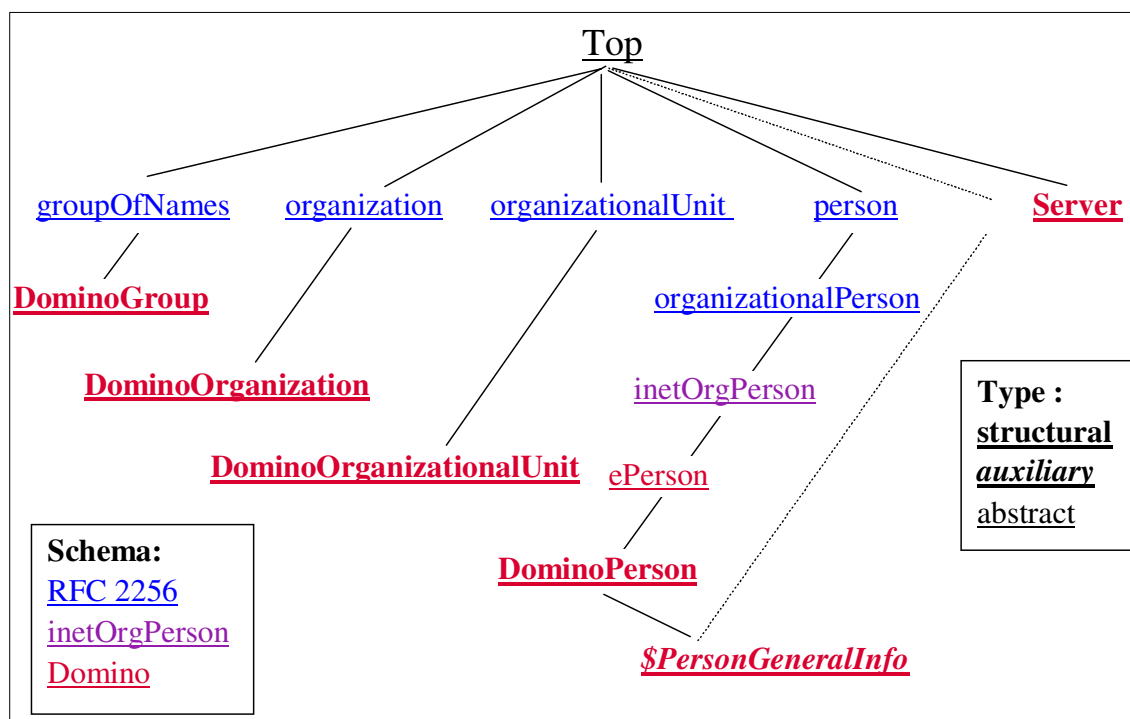


Figure 84. LDAP object classes in Domino R5.02

## H.1 Attributes

The attributes are listed in the following tables.

Table 10. General attributes

Generic Name	LIPS	Derivation	Comment
Electronic Mail	mail	inetOrgPerson	<i>rfc822mailbox</i> from RFC1274 is alias
Certificate	userCertificate	RFC2256	Notes defines two attributes: <i>userCertificate</i> & <i>certificate</i> (Domino)
Uniform Resource Locator	labeledURI	RFC2079	
Full Name	cn	RFC2256	Alias is <i>commonName</i>
Given Name	givenName	RFC2256	
Last Name	sn	RFC2256	Alias is <i>surName</i>

Generation Qualifier	generationQualifier	RFC2256	
Organization	o	RFC2256	Alias is <i>organizationName</i>
City	l	RFC2256	Alias is <i>localityName</i>
Country	c	RFC2256	Alias is <i>countryName</i>
Personal Title	personalTitle	inetOrgPerson	
Initials	initials	RFC2256	
Middle Name	middleName	Domino	
Unique Identifier	uniqueIdentifier	inetOrgPerson	Usually represented as <i>uid</i>

Table 11. Personal attributes

Generic Name	LIPS	Derivation	Comment
Home Telephone Number	homePhone	inetOrgPerson	RFC1274 element redefined
Home Fax	homeFax	Domino	
Home Postal Address	homePostalAddress	RFC1274	
Description	description	RFC2256	
Personal Photograph	thumbnailPhoto	N/A	inetOrgPerson <i>jpegPhoto</i> is preferred with <i>photo</i> as an alternative

Table 12. Organizational attributes

Generic Name	LIPS	Derivation	Comment
Title	title	RFC2256	
Office Telephone Number	telephoneNumber	RFC2256	
Office Fax Number	facsimileTelephoneNumber	RFC2256	
Office Mobile Telephone Number	mobileTelephoneNumber	inetOrgPerson	<i>mobile</i> is used; the LIPS attribute is the alias
Office Pager Number	pager	inetOrgPerson	<i>pagerTelephoneNumber</i> is an alias
Postal Address	postalAddress	RFC2256	
Organizational Department	ou	RFC2256	Alias is <i>organizationalUnit</i>
Room Number	physicalDeliveryOfficeName	RFC2256	inetOrgPerson considers this as Office Number as it introduces <i>roomNumber</i>
E-mail Address	textEncodedORaddress	inetOrgPerson	Used for X.400; alias <i>mhsORAddress</i>
Telex Telephone Number	telexNumber	RFC2256	
Company Logo	thumbnailLogo	N/A	inetOrgPerson <i>jpegPhoto</i> is preferred with <i>photo</i> as an alternative
Secretary	secretary	inetOrgPerson	RFC1274 element redefined
Manager	manager	inetOrgPerson	RFC1274 element redefined

Table 13. Security

Generic Name	LIPS	Derivation	Comment
User Password	userPassword	RFC2256	

Table 14. Ancillary

Generic Name	LIPS	Derivation	Comment
Creation Time	modifyTimeStamp		
Last Modified	modifyTimeStamp		
Creators Name	creatorsName	Domino	
Modifiers Name	modifiersName		

## H.2 Object classes and mapping to LDAP schema

The following three figures demonstrate the outline of the dominoGroup, dominoPerson and \$PersonGeneralInfo object classes, as well as the mapping onto the LDAP schema.

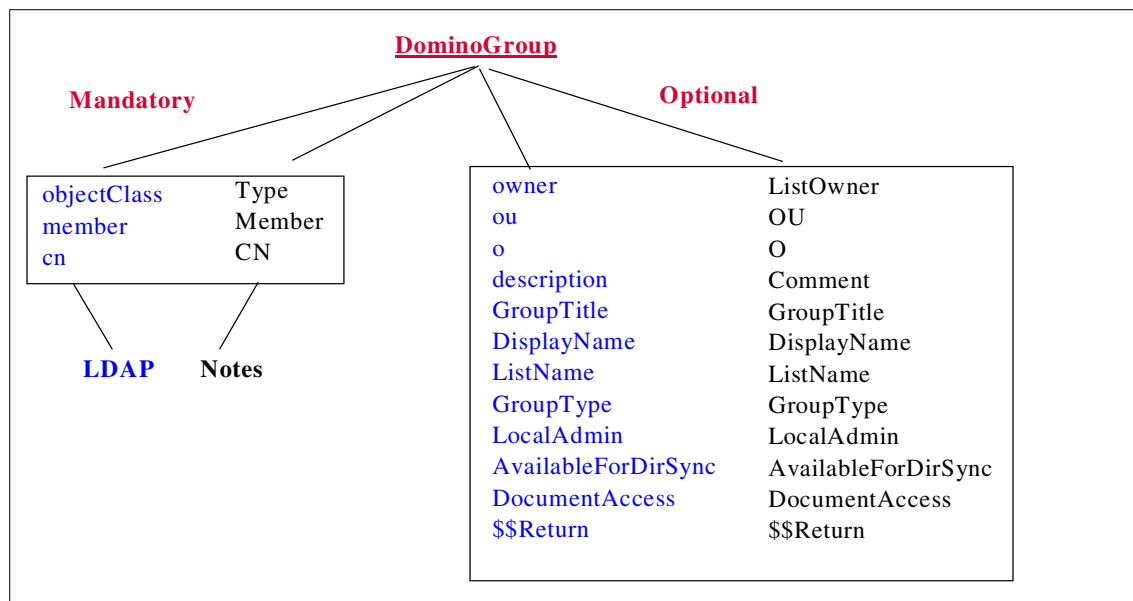


Figure 85. DominoGroup structural object class







full-search on the attribute, then look at the “Notes Name” column in the view.)

For each object class in the schema, the database provides the LDAP name, OID, Domino Directory form that corresponds to the object class, and LDAP schema the object class originated from. The database also provides the object class type (abstract, structural, auxiliary), the relative superior and auxiliary object classes, and the associated mandatory and optional attributes.

For each syntax, the database provides the LDAP name (and alternate name), OID, the data type mapping in Notes, and the schema the syntax originated from.

---

## Appendix I. Special notices

This publication is intended to help IT architects and Domino administrators to plan for and implement a directory structure using the Domino directory. The information in this publication is not intended as the specification of any programming interfaces that are provided by Lotus Domino. See the PUBLICATIONS section of the IBM Programming Announcement for Lotus Domino for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers

attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

3890	Lotus
400	Lotus Notes
Approach	Lotusphere
cc:Mail	XT
Domino	System/390
iNotes	WebSphere
Notes	
IBM ®	Redbooks
	Redbooks Logo 

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.



---

## Appendix J. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

---

### J.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 271.

- *Understanding LDAP*, SG24-4986
- *Lotus Notes and Domino R5.0 Security Infrastructure Revealed*, SG24-5341
- *LDAP Implementation Cookbook*, SG24-5110
- *Domino and WebSphere Together*, SG24-5955
- A Roadmap for Deploying Domino in the Organization, IBM form number SG24-5617, Lotus part number CT6P8NA
- The Three Steps to Super.Human.Software: Compare, Coexist, Migrate. From Microsoft Exchange to Lotus Domino. Part One: Comparison, IBM form number SG24-5614, Lotus part number CT7QTNA
- The Three Steps to Super.Human.Software: Compare, Coexist, Migrate. From Microsoft Exchange to Lotus Domino. Part Two: Coexistence and Migration, IBM form number SG24-5615, Lotus part number CT7QWNA
- Eight Steps to a Successful Messaging Migration: A Planning Guide for Migrating to Lotus Notes and Domino, IBM form number SG24-5335, Lotus part number CT6HINA
- The Next Generation in Messaging: Moving from Microsoft Mail to Lotus Notes and Domino, IBM form number SG24-5152, Lotus part number CT7SBNA
- The Next Generation in Messaging: Moving from Novell GroupWise to Lotus Notes and Domino, IBM form number SG24-5321, Lotus part number CT7NNNA
- Lotus Domino R5.0: A Developer's Handbook, IBM form number SG24-5331, Lotus part number CT6HPIE
- Lotus Notes and Domino R5.0 Security Infrastructure Revealed, IBM form number SG24-5341, Lotus part number CT6TPNA
- Enterprise-Wide Security Architecture and Solutions, IBM form number SG24-4579

- LotusScript for Visual Basic Programmers, IBM form number SG24-4856, Lotus part number 12498
- Secrets to Running Lotus Notes: The Decisions No One Tells You How to Make, IBM form number SG24-4875, Lotus part number AA0424
- Deploying Domino in an S/390 Environment, IBM form number SG24-2182, Lotus part number 12957
- Developing Web Applications Using Lotus Notes Designer for Domino 4.6, IBM form number SG24-2183, Lotus part number 12974
- High Availability and Scalability with Domino Clustering and Partitioning on Windows NT, IBM form number SG24-5141, Lotus part number CT6XMIE
- From Client/Server to Network Computing, A Migration to Domino, IBM form number SG24-5087, Lotus part number CT699NA
- AS/400 Electronic-Mail Capabilities, IBM form number SG24-4703
- Using Lotus Notes on the IBM Integrated PC Server for AS/400, IBM form number SG24-4779
- Managing Domino/Notes with Tivoli Manager for Domino, Enterprise Edition, Version 1.5, IBM form number SG24-2104
- Using ADSM to Back Up Lotus Notes, IBM form number SG24-4534
- NetFinity V5.0 Database Support, IBM form number SG24-4808
- Lotus Approach to DB2, IBM form number SG24-4685

---

## J.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at [ibm.com/redbooks](http://ibm.com/redbooks) for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694



---

### J.3 Other resources

These publications are also relevant as further information sources:

- e-Directories: Enterprise Software, Solutions, and Services, ISBN 0-201-70039-5

---

### J.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- <http://www.iit.edu/~gawojar/ldap/r>
- <http://www.notes.net/>
- <http://www.lotus.com/developer/>
- <http://java.sun.com/products/jndi/>
- <http://www.iit.edu/~gawojar/ldap>
- <http://www.notes.net>
- <http://www.lotus.com/developer>
- <http://java.sun.com/products/jndi>
- <http://www2.lotus.com/home.nsf/welcome/inotes>
- <http://www.innosoft.com>
- <http://www.openldap.org>
- <http://www.ic.siemens.com/networks/gg/isa/md/ps.htm>
- <http://modules.apache.org>
- <http://www.dsml.org>
- <http://www.dmtf.org>
- <http://www.opengroup.org>
- <http://www.directoryforum.org>
- <http://www.iso.ch>
- <http://www.itu.ch>
- <http://www.nexor.com>
- <http://www.ietf.org/rfc>
- <http://www.ietf.org/internet-drafts>
- <http://www.oasis.org>
- <http://www.redbooks.ibm.com>
- <http://www.architech.no>
- <http://www.novell.com>
- <http://w3.itso.ibm.com>

- <http://w3.ibm.com>
- <http://www.elink.ibm.link.ibm.com/pbl/pbl>

---

## How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** [ibm.com/redbooks](http://ibm.com/redbooks)

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the IBM Redbooks fax order form to:

	<b>e-mail address</b>
In United States or Canada	<a href="mailto:pubscan@us.ibm.com">pubscan@us.ibm.com</a>
Outside North America	Contact information is in the "How to Order" section at this site: <a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a>

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: <a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a>

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: <a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a>

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

---

### IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access [MyNews](http://w3.ibm.com/) at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

---

## IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

---

First name	Last name
------------	-----------

---

Company
---------

---

Address
---------

---

City	Postal code	Country
------	-------------	---------

---

Telephone number	Telefax number	VAT number
------------------	----------------	------------

---

<input type="checkbox"/> Invoice to customer number	
---	--

---

<input type="checkbox"/> Credit card number	
---	--

---

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**

---

## Index

### Symbols

\$CertifierExtensibleSchema 53  
\$DomainExtensibleSchema 53  
\$GroupExtensibleSchema 53  
\$MailInDatabaseExtensibleSchema 53  
\$PersonExtensibleSchema 53, 55  
\$ResourceExtensibleSchema 53

### A

Access Control List, see ACL  
ACL 80, 81, 125, 134, 222  
Active Directory 124  
address resolution 12, 15  
address verification 18  
Administrator  
    client 129  
administrator 9, 13, 69  
    database 26  
administrators 3  
ADSI 228  
aggregating 5, 60, 63, 65, 66, 77  
anonymous 7, 22, 50  
Apache Web Server 156  
API 218, 240  
application developers 3  
application integration 228  
Application Programming Interface, see API  
architecture 1, 5  
attribute mapping tables 227  
attributes 232  
authentication 114, 222  
    group 21  
    NTLM 132  
    Web 140  
    Web client 19  
author 26  
authorization 222  
availability 221

### B

base DN 58  
binding 240  
business drivers 33

### C

C language 219  
    API 240  
C/C++ 228  
Calendar Connector 47  
centralized 220  
certificate 6, 21  
    authority (CA) 23, 79  
chaining 236  
client  
    logon 123  
    non-Notes 7  
client/server model 218  
code reuse 3  
Common attributes 227  
ConsoleOne 142  
continuation reference 14  
cost of maintaining information 109

### D

DAP 230  
data validation 3  
database 223  
    administrator 26  
    validation logic 21  
DECS 47, 86  
depositor 26  
designer 26  
DIB 230  
dircat task 77  
directory 217  
    and databases 223  
    and transactions 224  
    architecture 5  
    distributed 220  
    foreign 121  
    integration 109  
    partitioned and replicated 220  
    profile 60, 66, 78  
    search order 15  
    security 221  
    server 75  
    servers and clients 218  
    services 6  
        markup language initiative 4  
        setup 47  
    synchronization 225

- services 116
  - telephone 217
- Directory Assistance 2, 5, 12, 56, 78, 79, 81, 86
  - configuration document 140
  - deploying 59
  - replicas 57
- Directory Catalog 2, 5, 9, 63, 78, 79, 86
  - configuration 12
  - mobile 10
    - deploying 66
  - multiple on workstation 68
  - server 11, 12
  - setup 59
  - Status Report agent 63
  - user 10
- Directory Information Tree 229
- Directory Interoperability Forum (DIF) 4
- DirX 144
- distinguished name 55, 77, 80
- distributed directory 220
- Distributed Management Task Force 4
- DIT 232
- DMD 231
- domain 7
- Domino
  - administrator 9
  - application server 47
  - Designer 87
  - directory services
    - setup 47
  - domain 7
  - Enterprise Server 47
  - Mail Server 47
- Domino Directory 1, 7
  - authentication 29
- DSA 230
- DUA 230

## E

- e-Business 3
- eDirectory 140
- editor 26
- encrypted mail 11
- enterprise directory 1, 41
- Entrust
  - PKI 156
- Eudora 31
- Exchange 116

- exhaustive lookup 78
- extensibleObject 242
- external LDAP directory 79
- external LDAP server 57

## F

- federation 5
- fewer name variations with higher security 78
- firewall 27, 221
- foreign directory 121
- full text index 60, 64, 65, 68, 78
- FullName 61, 77

## G

- global 220
- group
  - management 84
- group authentication 21
- group entries 125
- Group Expansion 58
- group types 62

## H

- hidden views 8
- HTTP
  - password 19, 78
- hub server 119

## I

- IBM SecureWay Directory 3
- IETF 4
- IMAP 29
- ImportLDIF 115
- Incremental fields 64
- indexed view 10, 62
- Innosoft
  - Distributed Directory Server (IDDS) 143
- iNotes 133
- integration directory 109
- Internet
  - Explorer 79
  - protocols 28
  - registration 118
- Internet Information Server (IIS) 116
- iPlanet 12, 134
  - directory server 134
  - smart referral 136

- web server 137
- ISAPI filter 131
- ISO
  - 9594 228

## J

- Java 228
  - application 219
- JDAP API 219
- JNDI 219, 228
- just in time encryption 62

## L

- LDAP 7, 29, 239
  - adding 48
  - API 219, 228
  - client authentication 27
  - compliant 12
  - directory 18
    - external 79
  - external server 57
  - history 228
  - lookups
    - troubleshooting 79
  - models 242
  - protocol or directory? 242
  - schema 52, 79
  - server task 5
  - service
    - loading 48
  - standards 228
  - support 47
  - Version 2 244
  - Version 3 110, 244
- LDAP\_Enforce\_Schema 87
- LDAPAddress 49
- ldapsearch 79
- LDAPSync 104, 115
- LDIF 139
  - commands 9
  - file 115
  - importing 133
- LEI 3.0 228
- Lightweight Directory Access Protocol (see also LDAP) 2
- ListName 61
- load LDAP 48
- local 220

- replica 68
- location document 75
- log\_dircat 70, 77
- logon
  - single 123
  - Windows client 123
- Lotus Domino Directory 3
- Lotus Professional Services 115

## M

- mail
  - address lookup 6
  - addressing 12
  - encrypted 11
  - file 118
- management agent 145
- manager 26
- maximum number of entries returned 51
- members 62
- merge factor 64
- metadirectory 227
- Microsoft 116
  - Metadirectory Services (MMS) 144
- Microsoft Exchange 116
- minimum characters for wildcard search 51
- mobile Directory Catalog 10
  - deploying 66
- MSFT 31
- multi-directory environment 110

## N

- Name and Address Book 1
- name resolution 6
- NDS 12
- nested groups 21
- Netscape 31
  - Communicator 79
  - iPlanet 12
  - Messenger 79
- network traffic 12
- NNTP 29
- No Access 26
- non-Notes client 7
- Notes
  - remote procedure call (NRPC) 23
  - short name 122
- notes.ini 48, 49
- Novell 140

- eDirectory 140
- NDS 12
- NRPC 7
- NTLM authentication 132

## O

- object 241
- object class 53, 125, 232, 241, 242
- ODBC 47, 86
- Open Group 4
- OpenLDAP 144
- OSI 228
- Outlook Express 79, 128
  - client 132

## P

- packet filtering 29
- packing density 64
- partitioning 220
  - server 49
- password
  - HTTP 19
- performance 59, 221
- PKI
  - Entrust 156
- POP3 29
- PostScript 218
- proxy 27
- public key infrastructure (PKI) 1

## R

- RDBMS 47
- RDN 233
- reader 26
- referral 13, 136, 220
  - services 7
- register person 120
- remote procedure call (RPC)
  - Notes 23
- replica 57, 68
  - copy 7
  - ID 11
- replication 64, 220
- requirements 33
- RFC 239
  - 1777 244
  - 2251 244

- RSA operations 26
- RunAgent 115

## S

- Schedule Manager 47
- schema 9, 109, 241
  - checking 87
  - LDAP 52
    - subclassing 241
  - schema50.nsf 52, 55, 56, 79
- search filter 240
- search order 15
- SecureWay 110
  - directory 3
- security 51, 85, 109, 221
  - authentication 222
  - authorization 222
- security policy 221
- selection formula 63
- selective replication 87
- server configuration document 49
- server Directory Catalog 12
- ServerTasks= 48
- short name 122
- show stat LDAP 71
- Siemens
  - DirX server 144
- single logon 123
- socks server 27
- soundex 12, 62
- SSL 29, 58, 81
  - protocol 58
- standards 239
- Structured Query Language (SQL) 225
- subclassing 241
- subforms 53
- SynchroNSF 115

## T

- telephone directory 217
- tell LDAP
  - exportschema 52, 55
  - reloadschema 55
- third-party 12
- timeout 51
- toolkit 2
- transaction 224
- trust 19



type ahead 6, 14, 62, 77, 135  
type down 14

## **U**

unbinding 240  
Unicode 51  
unique key 227  
updall 65  
user authentication 7  
user Directory Catalog 10  
users 3  
UTF8 51

## **V**

VeriSign 23  
view  
    indexed 10  
view indices 62  
views  
    hidden 8  
Visual Basic 228

## **W**

Web  
    authentication 140  
    server properties box 132  
WebAuth\_Verbose\_Trace 81  
WebSphere 114  
white pages 218  
wildcard 124  
Windows 2000 116  
Windows NT 116  
    directory synchronization services 116

## **X**

X.500 31, 81, 239

## **Y**

yellow pages 218

## **Z**

Zoomit 144



## IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook “made the difference” in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)

<b>Document Number</b>	SG24-5986-00
<b>Redbook Title</b>	Getting the Most From Your Domino Directory
<b>Review</b>	<div></div> <div></div> <div></div> <div></div> <div></div> <div></div>
<b>What other subjects would you like to see IBM Redbooks address?</b>	<div></div> <div></div> <div></div>
<b>Please rate your overall satisfaction:</b>	<input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor
<b>Please identify yourself as belonging to one of the following groups:</b>	<input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer <input type="radio"/> IBM, Lotus or Tivoli Employee <input type="radio"/> None of the above
<b>Your E-mail address:</b> The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	<input type="checkbox"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
<b>Questions about IBM's privacy policy?</b>	The following link explains how we protect your personal information. <a href="http://ibm.com/privacy/yourprivacy/">ibm.com/privacy/yourprivacy/</a>





**Getting the Most From Your Domino Directory**

(0.5" spine)

0.475" <-> 0.875"

250 <-> 459 pages







**Redbooks**

# Getting the Most From Your Domino Directory

**Using Domino  
Directory for a  
general purpose or  
enterprise directory**

**Using Directory  
Catalog and  
Directory Assistance**

**Integrating with  
other directories and  
applications**

From the very beginning, Domino Directory, originally known as the Name and Address Book (NAB), has been a key part of the Domino architecture. It has evolved from being specific to Lotus Notes and Domino to serving as a general purpose directory. This redbook explains Domino Directory services, how to plan for and implement them, and how they can be extended to work with other directory services.

We start by discussing the purpose of this redbook and the IBM-Lotus strategy for directories. Then we explore the wide range of services that Domino Directory offers. We also discuss access to Domino Directory from an application development angle through protocols and technologies such as Notes API, LDAP, and JNDI.

In addition, we look at technical approaches to directory integration and also a few real world examples where Domino Directory integrates with other directories. Finally, we cover some good practices in designing a directory infrastructure.

This redbook is written for IT architects, Domino administrators, and other technical professionals involved with directory structures.

**INTERNATIONAL  
TECHNICAL  
SUPPORT  
ORGANIZATION**

**BUILDING TECHNICAL  
INFORMATION BASED ON  
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)

SG24-5986-00

ISBN 0738419389